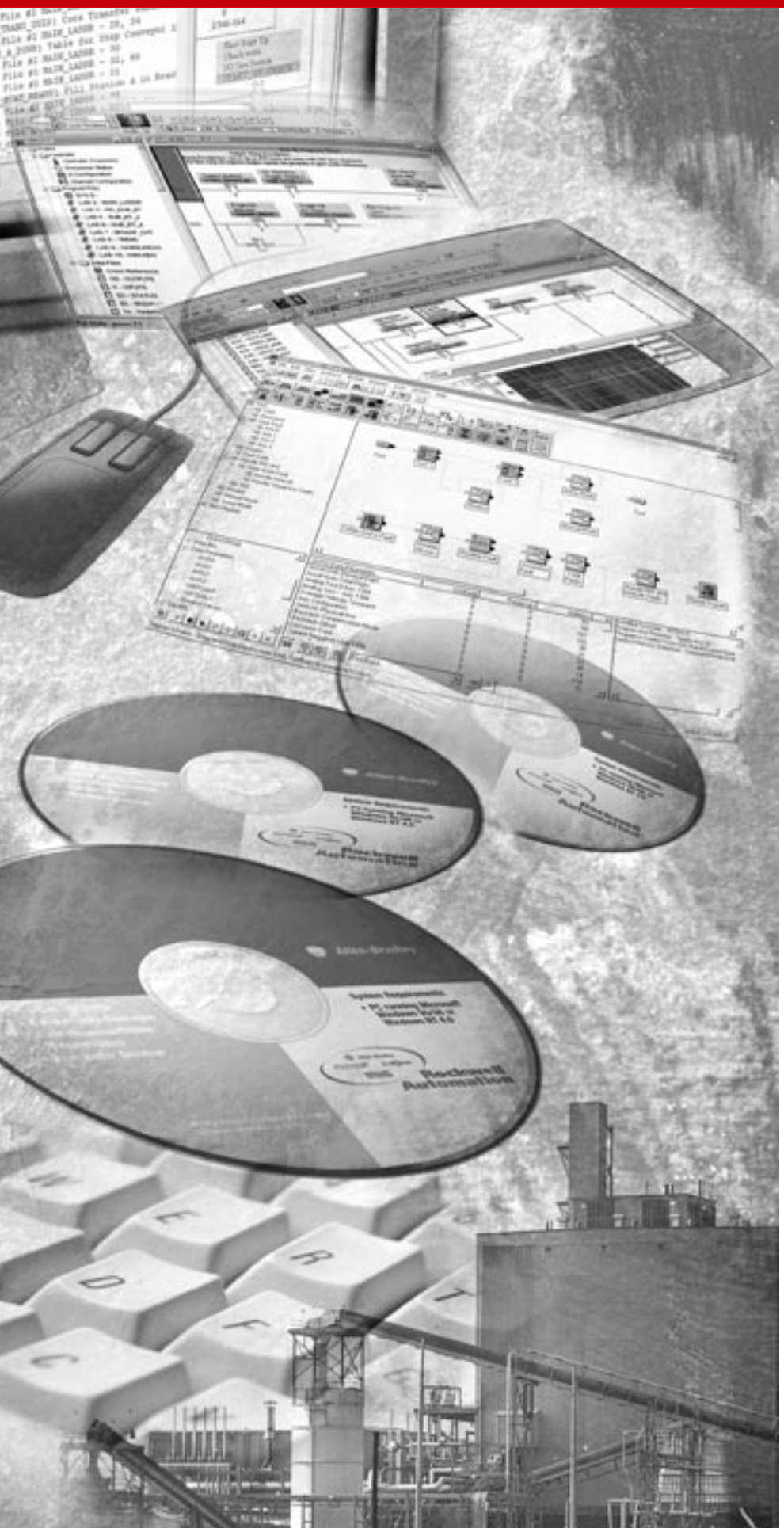


Allen-Bradley

DeviceLogix

User Manual

Rockwell Automation

Important User Information

Because of the variety of uses for the products described in this publication, those responsible for the application and use of this control equipment must satisfy themselves that all necessary steps have been taken to assure that each application and use meets all performance and safety requirements, including any applicable laws, regulations, codes and standards.

The illustrations, charts, sample programs and layout examples shown in this guide are intended solely for purposes of example. Since there are many variables and requirements associated with any particular installation, Allen-Bradley does not assume responsibility or liability (to include intellectual property liability) for actual use based upon the examples shown in this publication.

Allen-Bradley publication SGI-1.1, *Safety Guidelines for the Application, Installation and Maintenance of Solid-State Control* (available from your local Allen-Bradley office), describes some important differences between solid-state equipment and electromechanical devices that should be taken into consideration when applying products such as those described in this publication.

Reproduction of the contents of this copyrighted publication, in whole or part, without written permission of Rockwell Automation, is prohibited.

Throughout this manual we use notes to make you aware of safety considerations:

ATTENTION

Identifies information about practices or circumstances that can lead to personal injury or death, property damage or economic loss.

Attention statements help you to:

- identify a hazard
- avoid a hazard
- recognize the consequences

IMPORTANT

Identifies information that is critical for successful application and understanding of the product.

Allen-Bradley is a trademark of Rockwell Automation.
RSNetWorx for DeviceNet is a trademark of Rockwell Software, Inc.
DeviceNet is a trademark of Open DeviceNet Vendor Association.

European Communities (EC) Directive Compliance

If this product has the CE mark it is approved for installation within the European Union and EEA regions. It has been designed and tested to meet the following directives.

EMC Directive

This product is tested to meet the Council Directive 89/336/EC Electromagnetic Compatibility (EMC) by applying the following standards, in whole or in part, documented in a technical construction file:

- EN 50081-2 EMC — Generic Emission Standard, Part 2 — Industrial Environment
- EN 50082-2 EMC — Generic Immunity Standard, Part 2 — Industrial Environment

This product is intended for use in an industrial environment.

Low Voltage Directive

This product is tested to meet Council Directive 73/23/EEC Low Voltage, by applying the safety requirements of EN 61131-2 Programmable Controllers, Part 2 - Equipment Requirements and Tests. For specific information required by EN 61131-2, see the appropriate sections in this publication, as well as the Allen-Bradley publication Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1.

This equipment is classified as open equipment and must be mounted in an enclosure during operation to provide safety protection.

Notes:

Introduction — **Chapter 1**

What This Chapter Contains 1-1
What Is DeviceLogix? 1-1
DeviceLogix and RSNetWorx for DeviceNet 1-2
RSNetWorx for DeviceNet's Operating Modes 1-3
Terms, Acronyms, and Abbreviations 1-3
Audience 1-3
What We Assume 1-3
What's In This Manual? 1-4
Common Techniques Used In This Manual 1-4
Rockwell Automation Support 1-5
Local Product Support 1-5
Technical Product Assistance 1-5
Your Questions Or Comments About This Manual 1-5
What's Next? 1-5

Add Devices Off Line/On Line — **Chapter 2**

What This Chapter Contains 2-1
EDS Files 2-1
Unregister EDS files 2-1
Register EDS Files 2-5
Add Devices Off Line 2-9
Add Devices On Line 2-9
What's Next? 2-10

RSNetWorx for DeviceNet and the DeviceLogix Editor — **Chapter 3**

What This Chapter Contains 3-1
Access Device Properties 3-1
Check General Information 3-2
Enter Device Parameters 3-4
Understand Parameters 3-6
Access Default I/O Information 3-16
Access EDS Information 3-17
Access DeviceLogix-Enabled Devices 3-18
Use DeviceLogix Editor Tools 3-19
Create Logic Using the Schematic Sheet 3-20
Use Menus Within the DeviceLogix Editor. 3-20
Display Toolbars Within the DeviceLogix Editor 3-27
Describe the Status Bar Within the DeviceLogix Editor . 3-33
What's Next? 3-34

DeviceLogix Components

Chapter 4

What This Chapter Contains 4-1
About DeviceLogix Components 4-1
Create Logic With Function Blocks. 4-1
Attach Inputs and Outputs To Function Blocks 4-2
Configure With Function Blocks 4-2
Create Logic With Inputs 4-10
About Hardware Inputs 4-10
About Network Inputs 4-10
About Status Inputs 4-11
About Faults 4-11
Create Logic With Outputs. 4-11
About Hardware Outputs 4-11
About Network Outputs 4-11
What's Next? 4-12

Create Logix Using the DeviceLogix Editor

Chapter 5

What This Chapter Contains 5-1
Create a Schematic 5-1
Add Elements To the Schematic 5-1
Select an Element Other Than a Function Block 5-3
Place an Element Into the Schematic. 5-4
Make Connections 5-5
Delete a Wire: 5-6
Determine the State Of a Condition 5-6
Negate Data. 5-6
Set Assume Data Available 5-7
Interface a DeviceLogix Device To a Scanner 5-8
Interface to a DeviceLogix Device Without a Network 5-10
Edit On Line 5-10
Edit While On Line 5-11
Troubleshoot and Debug. 5-12
Forcing 5-12
Force Inputs 5-13
Force Outputs 5-13

Introduction

What This Chapter Contains

This chapter introduces you to DeviceLogix. The following table lists what this chapter contains and where to find specific information.

For information about	See page
what is DeviceLogix	1-1
terms, acronyms, and abbreviations	1-3
audience	1-3
what we assume	1-3
what's in this manual	1-4
common techniques used in this manual	1-4
Rockwell Automation support	1-5
What's next	1-5

What Is DeviceLogix?

DeviceLogix is a function that has been added to a number of Rockwell Automation devices to control outputs and manage status information locally within the device. This is accomplished through the DeviceLogix Editor.

The DeviceLogix Editor is the configuration tool for DeviceLogix-capable devices. It is a software tool that provides a graphical interface for configuring Function Blocks to provide local control within DeviceLogix-capable devices. The tool is an applet of RSNetWorx for DeviceNet™ which means that DeviceLogix is launched from within RSNetWorx for DeviceNet. The DeviceLogix Editor works with all DeviceLogix-capable devices. This manual tells you how to apply DeviceLogix within the products that support this function.

A DeviceLogix-capable device operates as a slave device on a DeviceNet™ network. You can enable a logic operation using the DeviceLogix Editor to provide local control over outputs on a device. The DeviceLogix device consists of:

- a specific number of inputs and/or outputs
- local logic that determines its behavior

Inputs and Outputs

Inputs and outputs can be one of two types:

- Physical - Inputs and outputs realized by physical connections to the device. These are referred to as Discrete Inputs or Discrete Outputs.
- Networked - Inputs consumed by the device from the network and outputs produced by the device onto the network.

Local Logic

The local logic of a DeviceLogix device consists of Function Blocks, inputs, outputs, and connections (wires) between them. Function Blocks contain connection points (hereafter called pins) and perform a specific function. Inputs and outputs also have connection pins and represent the actual hardware devices, networked data, and fault and status bits that are available for use in the local logic.

A connection (hereafter call wire) between Function Blocks is defined when an input pin of one Function Block is bound to an output pin of another Function Block. A pin can be bound to a:

- pin of another Function Block
- physical input
- networked input
- fault or status bit

Function Blocks may also have attributes that influence their function.

Configuring a DeviceLogix device consists of defining or editing the local logic that is present on the device along with the EDS parameters for that device.

DeviceLogix and RSNetWorx for DeviceNet

You will configure DeviceLogix features through RSNetWorx for DeviceNet. The DeviceLogix Editor ships as part of RSNetworx for DeviceNet starting with revision 3.0. EDS files that enable DeviceLogix are shipped with the release of RSNetworx in a separate folder labeled **Additional EDS Files**.

RSNetWorx for DeviceNet's Operating Modes

RSNetWorx for DeviceNet operates in one of two modes:

- offline - RSNetWorx for DeviceNet is not connected to the network.
- online - RSNetWorx for DeviceNet is connected to the network and is capable of communicating with devices on the network.

RSNetWorx for DeviceNet lets you select online or offline mode. Whichever mode RSNetWorx for DeviceNet is in directly affects the way DeviceLogix behaves when it is launched.

Terms, Acronyms, and Abbreviations

Refer to the following table to become familiar with DeviceLogix.

Name	Description
applet	an application program that uses the client to provide a user interface.
download	transfer logic from the software memory to the device.
logic	logic consisting of Function Blocks and their interconnections that can reside on a DeviceLogix device.
MAC ID	Media Access Control Identifier - An integer identification value assigned to each node on DeviceNet. This value distinguishes a node among all other nodes on the same link.
upload	transfer logic from the device memory to the software memory.

Audience

This manual is intended for engineers and technicians who use DeviceLogix to control outputs and manage information locally within devices.

What We Assume

We assume you are familiar with the working environment of RSNetWorx for DeviceNet and with configuring distributed I/O devices.

What's In This Manual?

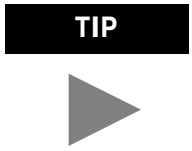
In this manual, we describe how the DeviceLogix Editor interacts with RSNetworx for DeviceNet then we describe the specifics of configuring DeviceLogix. In this manual:

To learn how to	See
add devices on line and off line	Chapter 2
access RSNetWorx for DeviceNet and DeviceLogix general features	Chapter 3
use DeviceLogix components	Chapter 4
create logic using the DeviceLogix Editor	Chapter 5

Common Techniques Used In This Manual

The following conventions are used throughout this manual:

- Bulleted lists provide information, not procedural steps.
- Numbered lists provide sequential steps.
- Pictures of keys and/or screens represent the actual keys you press or the screens you use.
- Actions you must perform appear in bold and look like the following example. Select **Unregister a device**.
- a menu item in this format **Network>Online** identifies the menu item (**Network)** and the submenu item (**Online)** after the caret (>).
- We sometimes use the word 'editor' to refer to the DeviceLogix Editor.
- We sometimes use RSNetWorx to refer to RSNetWorx for DeviceNet.



Tips contain helpful information.

Rockwell Automation Support

Rockwell Automation offers support services worldwide, with over 75 sales/support offices, 512 authorized distributors, and 260 authorized systems integrators located throughout the United States alone, plus Rockwell Automation representatives in every major country in the world.

Local Product Support

Contact your local Rockwell Automation representative for:

- sales and order support
- product technical training
- warranty support
- support service agreements

Technical Product Assistance

If you need to contact Rockwell Automation for technical assistance, call your local Rockwell Automation representative.

Your Questions Or Comments About This Manual

If you find a problem with this manual, please notify us of it on the enclosed Publication Problem Report (at the back of this manual).

If you have any suggestions about how we can make this manual more useful to you, please contact us at the following address:

Rockwell Automation, Allen-Bradley Company, Inc.
Automation Control and Information Group
Technical Communication
1 Allen-Bradley Drive
Mayfield Heights, OH 44124-6118

What's Next?

Now that you have been introduced to DeviceLogix, read Chapter 2 to learn how to register EDS files and how to add devices on line or off line.

Notes:

Add Devices Off Line/On Line

What This Chapter Contains

Read this chapter to learn how to add devices to the network on line or off line. In this chapter you will learn how:

To	See page
unregister and register EDS files	2-1
add devices off line	2-9
add devices on line	2-9

EDS Files

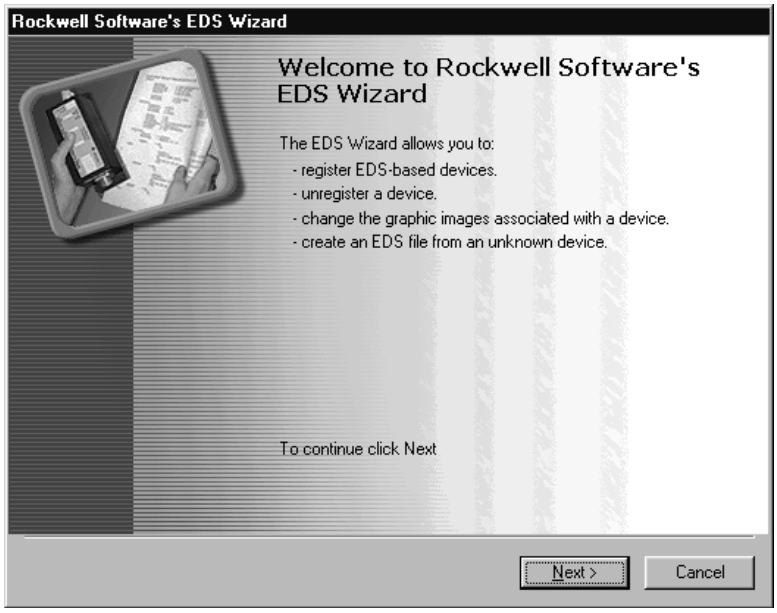
Before you start to add devices, you must register the EDS file for the target device. Use the EDS Wizard in RSNetWorx for DeviceNet to register the EDS file. For best results, you should unregister the existing EDS file before registering the new EDS file.

Unregister EDS files

To unregister an existing EDS file:

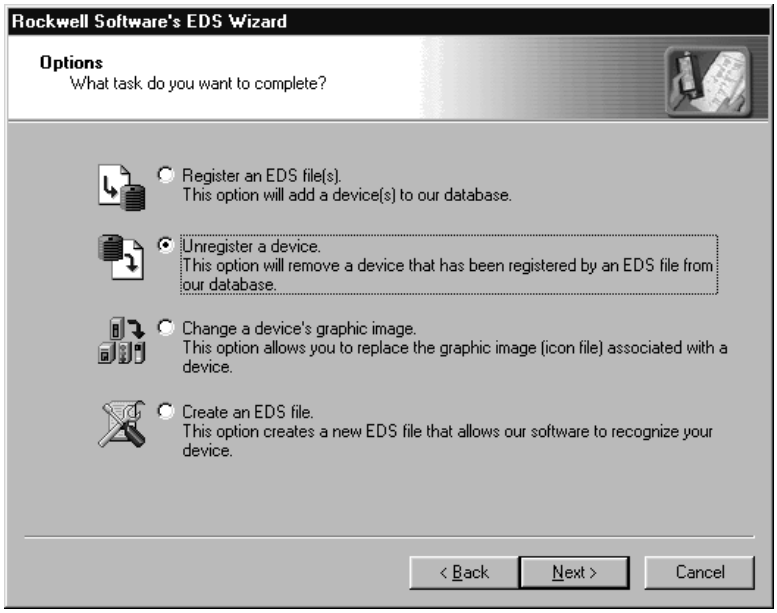
1. Open RSNetWorx for DeviceNet.
2. Select **Tools>EDS Wizard**.

   You see the welcome window for the EDS Wizard.

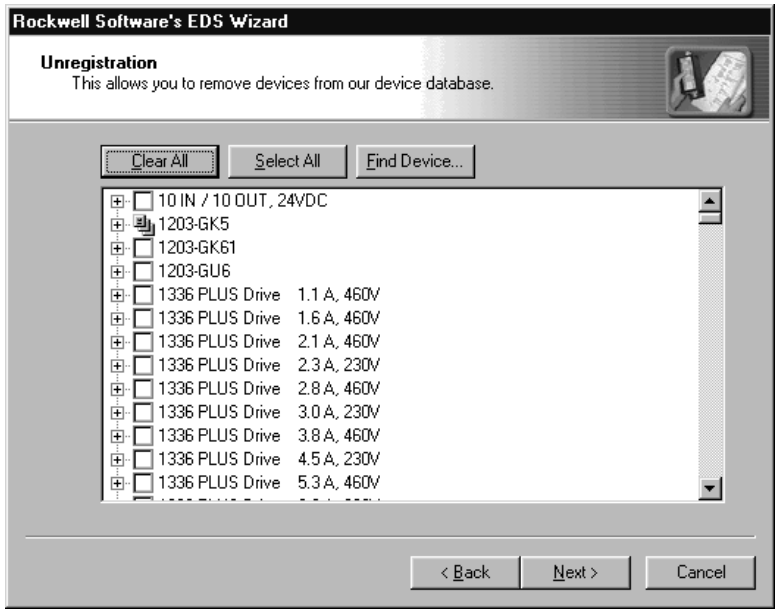

Rockwell Software's EDS Wizard
Welcome to Rockwell Software's EDS Wizard
The EDS Wizard allows you to:
- register EDS-based devices.
- unregister a device.
- change the graphic images associated with a device.
- create an EDS file from an unknown device.
To continue click Next
Next >
Cancel

3. Click **Next** to start.

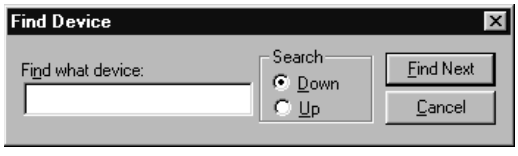
4. Select **Unregister a device**.


Rockwell Software's EDS Wizard
Options
What task do you want to complete?
Register an EDS file(s).
This option will add a device(s) to our database.
Unregister a device.
This option will remove a device that has been registered by an EDS file from our database.
Change a device's graphic image.
This option allows you to replace the graphic image (icon file) associated with a device.
Create an EDS file.
This option creates a new EDS file that allows our software to recognize your device.
< Back
Next >
Cancel

5. Click **Next**.


Rockwell Software's EDS Wizard
Unregistration
This allows you to remove devices from our device database.
Clear All
Select All
Find Device...
10 IN / 10 OUT, 24VDC
1203-GK5
1203-GK61
1203-GU6
1336 PLUS Drive 1.1 A, 460V
1336 PLUS Drive 1.6 A, 460V
1336 PLUS Drive 2.1 A, 460V
1336 PLUS Drive 2.3 A, 230V
1336 PLUS Drive 2.8 A, 460V
1336 PLUS Drive 3.0 A, 230V
1336 PLUS Drive 3.8 A, 460V
1336 PLUS Drive 4.5 A, 230V
1336 PLUS Drive 5.3 A, 460V
< Back
Next >
Cancel

6. Click **Find Device**.

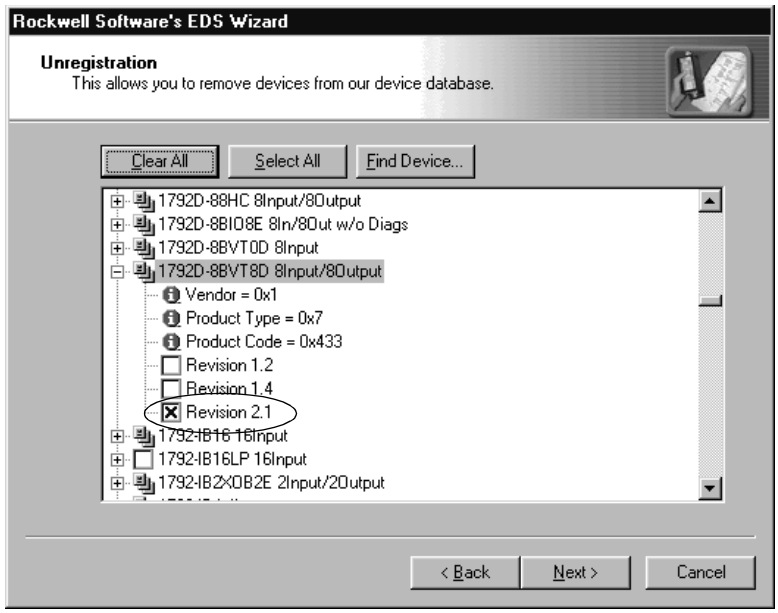


7. Type the name of the device you want to unregister in the **Find what device** box.

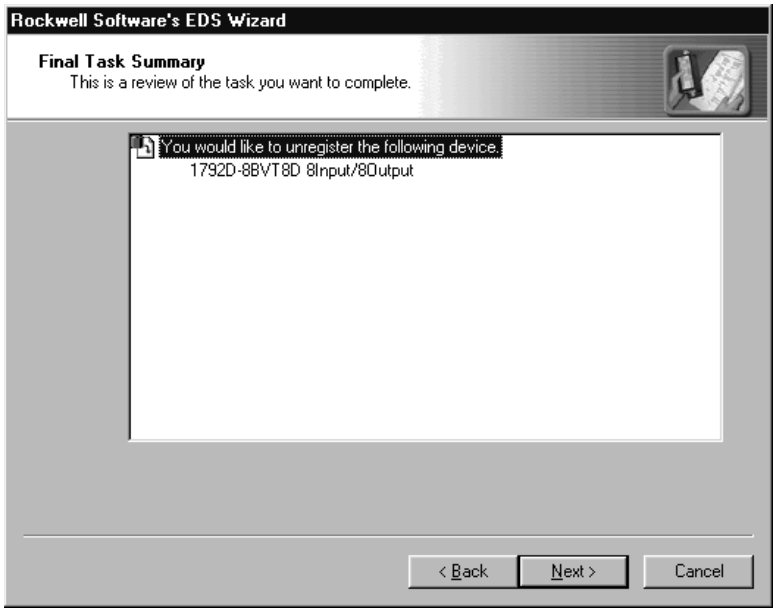
8. Click **Find Now**.

9. When the device appears in the EDS Wizard window, click **Cancel**.

10. Click the box in front of the device or revision so an **x** appears.



11. Click **Next**.


Rockwell Software's EDS Wizard
Final Task Summary
This is a review of the task you want to complete.
You would like to unregister the following device.
1792D-8BVT8D 8Input/8Output
< Back
Next >
Cancel

12. Click **Next** to unregister the EDS file.

Rockwell Software's EDS Wizard
Completing the EDS Wizard
You have successfully completed the EDS Wizard.
The changes you have made may not show up until all RSI applications are shut down.
Finish

You have successfully unregister the existing EDS file.

13. Click **Finish** to close the EDS Wizard.

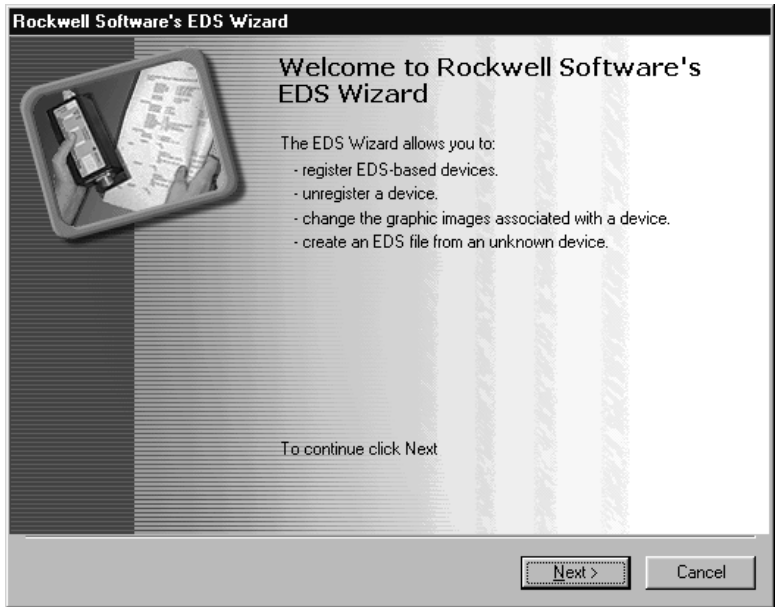
14. Close RSNetWorx for DeviceNet and reopen it for the change to take effect.

Register EDS Files

To register EDS files:

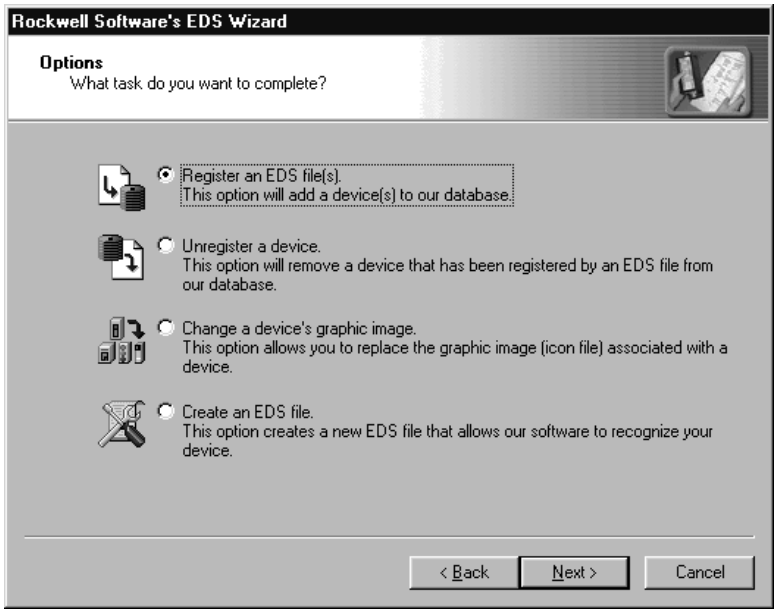
1. Open RSNetWorx for DeviceNet, if it is not already open.
2. Select **Tools>EDS Wizard**.

   You see the welcome window for the EDS Wizard.


Rockwell Software's EDS Wizard
Welcome to Rockwell Software's EDS Wizard
The EDS Wizard allows you to:
- register EDS-based devices.
- unregister a device.
- change the graphic images associated with a device.
- create an EDS file from an unknown device.
To continue click Next
Next >
Cancel

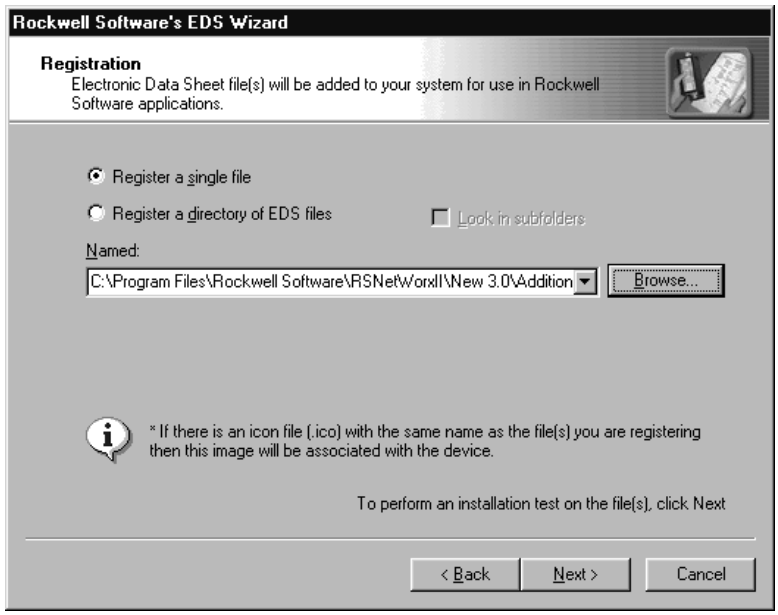
3. Click **Next** to start.

4. Verify that **Register an EDS file(s)** is selected.


Rockwell Software's EDS Wizard
Options
What task do you want to complete?
Register an EDS file(s).
This option will add a device(s) to our database.
Unregister a device.
This option will remove a device that has been registered by an EDS file from our database.
Change a device's graphic image.
This option allows you to replace the graphic image (icon file) associated with a device.
Create an EDS file.
This option creates a new EDS file that allows our software to recognize your device.
< Back
Next >
Cancel

5. Click **Next**.

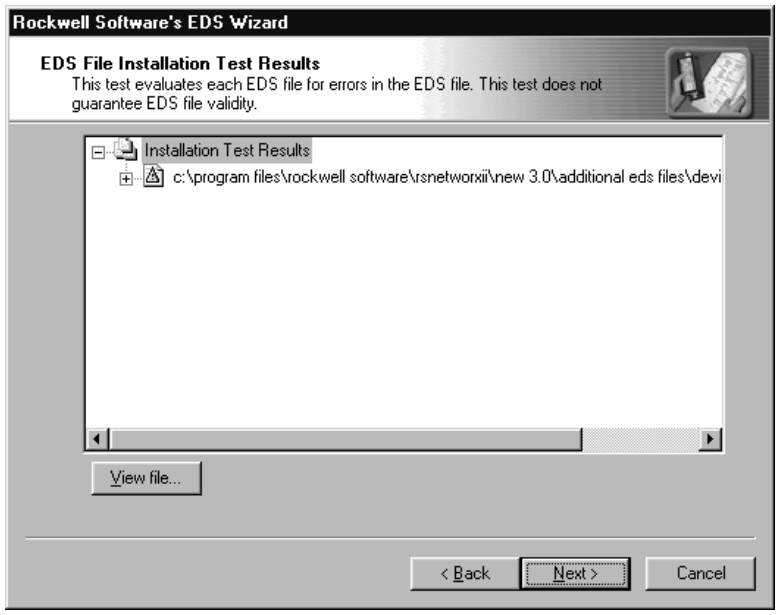
6. Choose to register a single file or a directory of files.


Rockwell Software's EDS Wizard
Registration
Electronic Data Sheet file(s) will be added to your system for use in Rockwell Software applications.
Register a single file
Register a directory of EDS files
Look in subfolders
Named:
C:\Program Files\Rockwell Software\RSNetWorxII\New 3.0\Addition
Browse...
* If there is an icon file (.ico) with the same name as the file(s) you are registering then this image will be associated with the device.
To perform an installation test on the file(s), click Next
< Back
Next >
Cancel

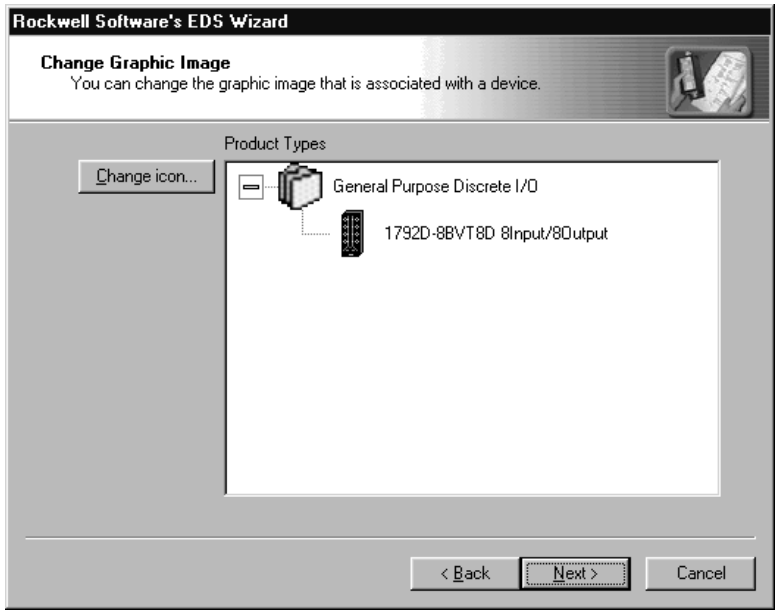
7. Click **Browse** if you need to find the correct file.

8. Click **Next**.

9. Click **Next** to evaluate the EDS files for errors.

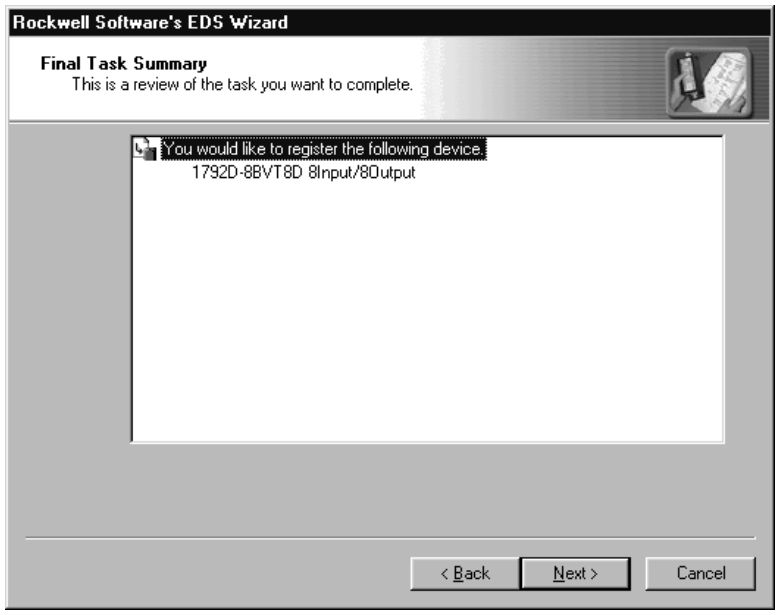

Rockwell Software's EDS Wizard
EDS File Installation Test Results
This test evaluates each EDS file for errors in the EDS file. This test does not guarantee EDS file validity.
Installation Test Results
c:\program files\rockwell software\rsnetworxii\new 3.0\additional eds files\devi
View file...
< Back
Next >
Cancel

10. (optional) Click **Change icon** to choose a different icon.



11. Click **Next**.

12. Verify that you are registering EDS files.


Rockwell Software's EDS Wizard
Final Task Summary
This is a review of the task you want to complete.
You would like to register the following device.
1792D-8BVT8D 8Input/8Output
< Back
Next >
Cancel

13. Click **Next**.

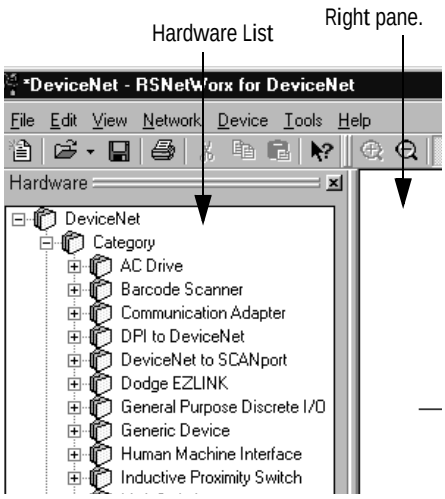
You have successfully registered the EDS file(s).

14. Click **Finished**.

15. Close RSNetWorx for DeviceNet and reopen it for the changes to take effect. If you continue to have problems registering EDS files, open RSNetWorx for DeviceNet and follow this path for additional help: **Help>Release Notes>Known anomalies>Devices unrecognized after registering EDS files with the EDS Wizard**.

Add Devices Off Line

Once the EDS files are registered, you can add and configure devices. If you are not connected to a DeviceNet network, you can add devices off line. Follow the steps below to add devices off line.



1. Start RSNetWorx for DeviceNet, if it is not already open.
2. Once RSNetWorx for DeviceNet is open, search the hardware list for the desired category.
3. Click the **+** sign in front of the desired category to expand the list.
4. Search the list of products for the device you want to add.
5. Click once on the device you want to add and hold down the mouse key.
6. Drag and drop the device onto the graph (right pane).
7. Add other devices as needed by following steps 3 through 6 above.

Add Devices On Line

If you are connected to a DeviceNet network, you can add devices on line. Before you attempt to go on line, make sure all the devices on the network:

- have a unique DeviceNet address
- are powered

Once you have accomplished the above tasks, access RSNetWorx for DeviceNet and select **Network>Online**.

RSNetWorx for DeviceNet scans the DeviceNet network and adds all devices it finds. It also finds the appropriate EDS file definition for the revision of the device.

You are now on line.

If the device is DeviceLogix compliant and the EDS file has been properly imported into RSNetWorx for DeviceNet, the device is ready to configure.

If a problem exists, a symbol appears above the device icon as displayed in RSNetWorx for DeviceNet. See the appropriate documentation for RSNetWorx for DeviceNet for details on how to resolve discrepancies that occur.

What's Next?

You now know how to register EDS files and how to add devices on line or off line. Read Chapter 3 to access the RSNetWorx for DeviceNet device properties dialog box and the DeviceLogix Editor to set up general properties, determine parameters and use DeviceLogix Editor tools.

RSNetWorx for DeviceNet and the DeviceLogix Editor

What This Chapter Contains

In this chapter you will access the RSNetWorx for DeviceNet device properties dialog box and the DeviceLogix Editor to set up general properties, determine parameters and use DeviceLogix Editor tools.

Read this chapter to	See page
check general properties	3-2
enter device parameters	3-4
understand parameters	3-6
determine parameters	3-10
access I/O defaults	3-16
access EDS Files	3-17
access DeviceLogix-enabled devices	3-18
create logic using the schematic sheet	3-20
use menus within the DeviceLogix Editor	3-20
display toolbars within the DeviceLogix Editor	3-27
describe the status bar within the DeviceLogix Editor	3-33

Access Device Properties

Access device properties and DeviceLogix features from the device properties dialog box in RSNetWorx for DeviceNet. The device properties dialog box is the same for both DeviceLogix devices and non-DeviceLogix devices, except when you are working with a DeviceLogix enabled device, you see an additional tab called **DeviceLogix** in the properties dialog box.

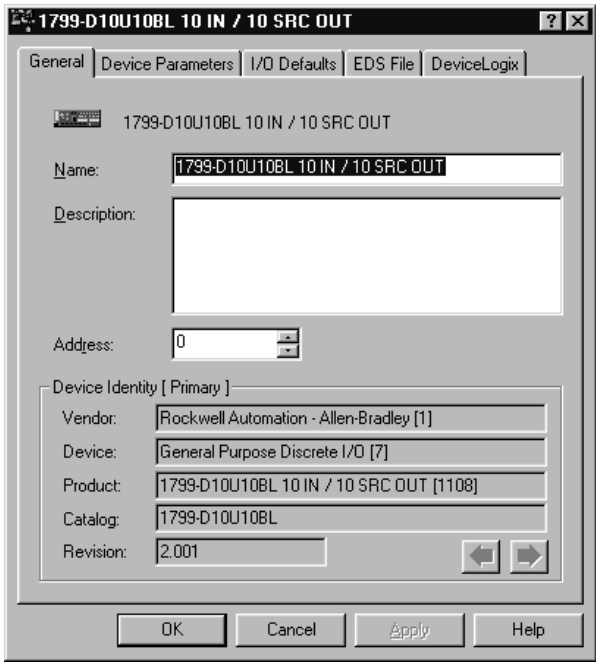
To access the device properties dialog box:

- Double-click the DeviceLogix-enabled device you added on the RSNetWorx for DeviceNet graph (right pane).

The following sections describe the tasks you perform in the device properties dialog box that affect DeviceLogix features. For a complete discussion of the general features of the RSNetWorx for DeviceNet device properties dialog box, see the RSNetWorx for DeviceNet documentation.

Check General Information

The device properties dialog box opens to the **General** properties window. If you are in another window in the device properties dialog box, click on the **General** tab to return to the **General** window.



The **General** window is common to both DeviceLogix devices and non-DeviceLogix devices. For DeviceLogix devices, you should make special note of the **Address** and **Revision** fields.

Set the Node Address

The address in this window must match the physical address on the module. If you added your devices off line, you most likely need to adjust the addresses of the devices. If you need to adjust the address:

1. Click once in the **Address** box.

2. Change the number of the address so that it matches the physical address on the module.

3. Click **OK**.

   If you added the devices on line by letting RSNetWorx for DeviceNet scan the network for devices, then the addresses in the General window should already match the physical addresses of the devices.

Check Revision Field

The revision field shows the current revision of the firmware in the I/O device. The revision field can help you identify if you are using the correct EDS file for DeviceLogix devices. If you are not using the correct EDS file for DeviceLogix devices, DeviceLogix features will not be enabled. See the **Access EDS Information** section, page 3-17, for important information about DeviceLogix information in the EDS file.

Using Window Buttons

The buttons at the bottom of the **General** window are common to all of the windows in the device properties dialog box. These buttons are described below.

- **OK**
  Saves changes and closes the device properties dialog box
- **Cancel**
  Discards changes made after the last time the **Apply** or **OK** button was used
- **Apply**
  Saves changes without closing the device properties dialog box
- **Help**
  Displays help for the current window

IMPORTANT Even though the **OK** and **Apply** buttons save changes made in the device properties dialog box, they do not actually save the information to the DNT file or make changes to the configuration stored in the I/O device. You can only save information to the DNT file by using the **Save** option in RSNetWorx for DeviceNet.

If you make changes while on line, you will be prompted to download or upload the device information whenever you try to go to a different window in the device properties dialog box. Press **OK** or **Apply** at the prompt. If you make changes while off line, you will not be prompted to upload or download the device until you go on line and try to go to a different window in the device properties dialog box.

Once you have entered the desired information in the **General** window, you can click on other tabs on the device properties dialog box.

IMPORTANT If you are on line and you click on either the **Device Parameters** or the **DeviceLogix** tab, you may be prompted to upload or download the device. When you are on line, the dialog checks the configuration in the device and compares it to the current configuration. If the configurations are not the same, you must upload or download the device to make the configurations the same before you can make changes. If you need to make changes without uploading or downloading, you can exit the dialog box, go off line and reenter the dialog box to make the desired changes.

Enter Device Parameters

In this section, we describe the general parameters of the **Device Parameters** window and help you become familiar with the parameters specific to DeviceLogix. We do not describe parameters that are device specific. The only parameters discussed in detail are those that affect DeviceLogix features. For more information on other parameters presented in this window, please refer to documentation that comes with your specific device.

To access the **Device Parameters** window, click on the **Device Parameters** tab.

Device Parameter Controls

There are a number of controls presented in the **Device Parameters** window. Most of the controls are available both on line and off line, while others are available only on line. The on line only features include an **Upload**, **Download**, and **Start Monitor** button as described below.

- **Upload**
  Reads the configuration data from the device.
- **Download**
  Writes the current configuration to the device.

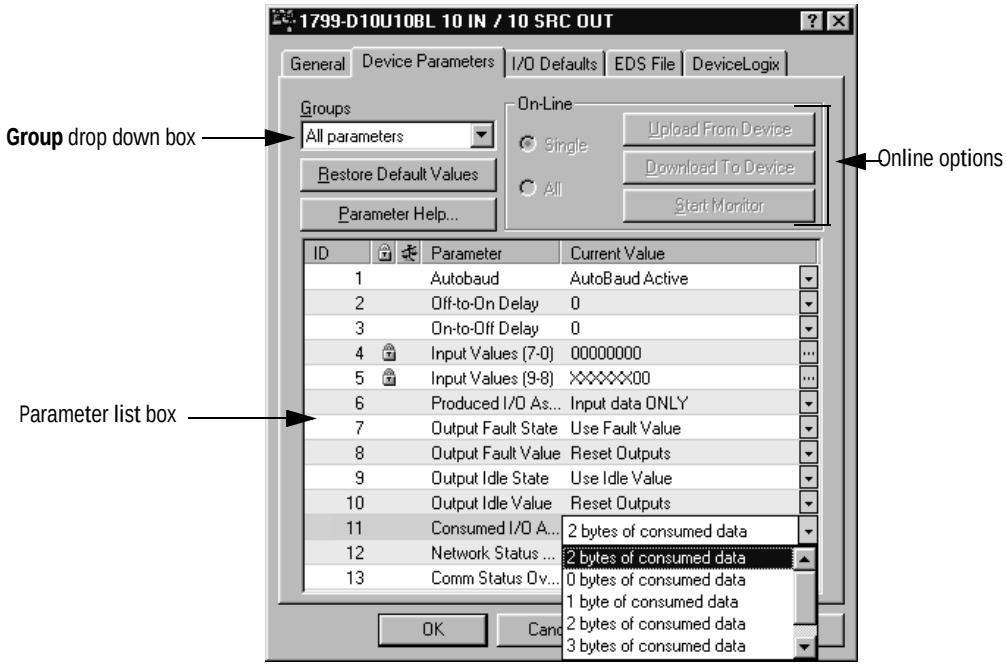
- **Start Monitor**

  Continually monitors either a single parameter in the parameter list or all the parameters in the parameter list. You tell the monitor routine whether you want to monitor a single parameter or all parameters using the **Single** or **All** buttons located to the left of the on-line buttons. Note that when you click the **Start Monitor** button, the button name changes to **Stop Monitor**. You can toggle the monitor mode on or off by clicking this button.

Features available for both on line and off line use include controls that let you: select which parameters to view, restore parameter defaults, access help for specific parameters, and access a list of parameters and their current values.

Select Group Parameters

The **Group** box drop down list lets you select which parameters you wish to view in the parameter list. The default is all the parameters. The options available in the drop down list varies based on the product. Click on the arrow to the right of the drop down box to select another group.



Restore Default Values

The **Restore Default Values** button resets all the parameters to their default values as defined in the EDS file. The default values could be different from the values last saved in the .DNT file. The default values may also be different from the values stored in the device. The default values are in the device when it is shipped from the factory. To restore default values:

1. Click the **Restore Default Values** button.
2. Click **Yes** to restore default values.

Access Parameter Information

Use the **Parameter Help** button to display a short description of a parameter.

1. Click the parameter in the list box at the bottom of the window.
2. Click the **Parameter Help** button to see the short description.

Understand Parameters

The bottom portion of the **Device Parameters** window contains the parameter list box. All the parameters of the I/O device that can be modified are listed here. Most of these parameters are specific to the I/O device and will not be discussed in this publication. However, the following four parameters are specific to DeviceLogix:

- Comm Status Override
- Network Status Override
- Consumed I/O Assembly
- Produced I/O Assembly

Before we help you determine values for DeviceLogix parameters, we will discuss network I/O.

Understand Network I/O

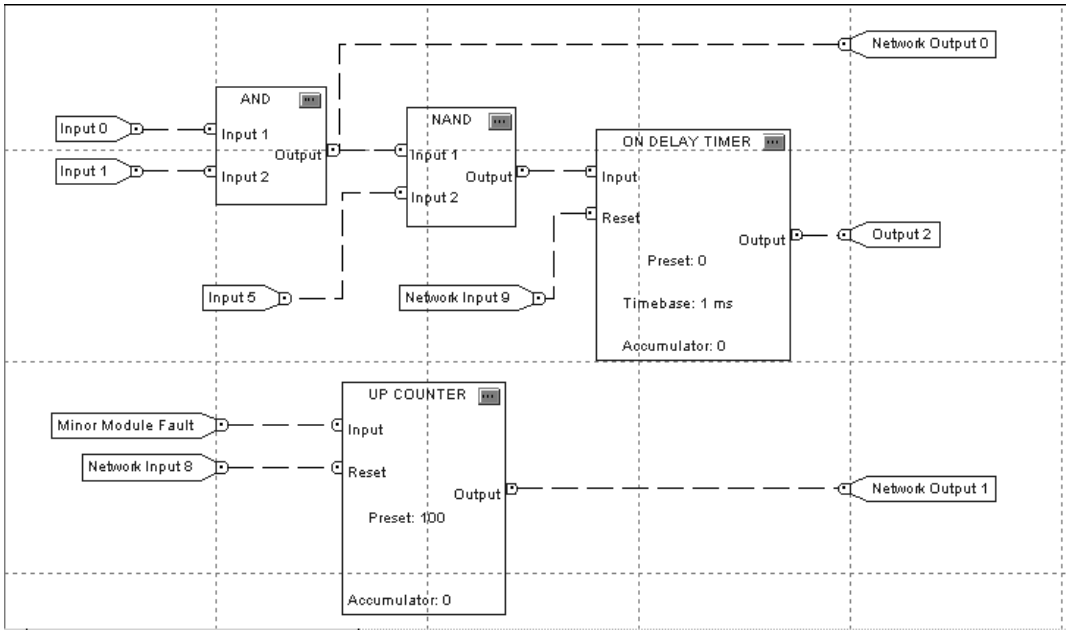
Network I/O is data that is consumed or produced by a DeviceLogix-capable device that is not directly associated with any hardware on the device. The produced and consumed cases will be discussed separately.

Understand Produced Network I/O (also known as Network Outputs)

Under normal conditions, an I/O device will produce the state of its inputs and the status of any fault information on the device. However with local logic running on a device, a master controller may need to know the results of some intermediate state of logic. Using a special I/O assembly containing Network Outputs, the device can report the state of any portion of the logic.

Each Network Output has a bit reserved for it in the module's produced I/O assembly. When you connect this Network Output to something in the logic, those results will be reported in the produced data.

In the example below, the user wishes to know when hardware inputs 0 and 1 are on at the same time. This will be reported in Network Output 0. Network Output 1 will report when 100 Minor Module Faults have occurred. Both of these pieces of information are reported in the produced I/O data shown below the example.



	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	In 7	In 6	In 5	In 4	In 3	In 2	In 1	In 0
Byte 1		OPWR	Logic Ena					
Byte 2	Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0
Byte 3	Network Output 7	Network Output 6	Network Output 5	Network Output 4	Network Output 3	Network Output 2	Network Output 1	Network Output 0

All data in the assembly including Network Outputs are capable of causing a Change of State production. OPWR = Output Power; Logic Enabled = DeviceLogix Enabled

Understand Consumed Network I/O (also known as Network Inputs)

Under normal conditions, an I/O device would consume data to apply to its hardware outputs. DeviceLogix-based devices may consume additional information to use in local logic.

A DeviceNet assembly is a collection of parameters from one or more objects. The consumed I/O assembly for a device with digital outputs contains the “Value” parameters of the Discrete Output Point Objects representing the hardware outputs present on the device. In other words, the assembly is sent with the data to control the hardware outputs on the device. Such an assembly for a module with 8 digital outputs is shown below.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Output Value 7	Output Value 6	Output Value 5	Output Value 4	Output Value 3	Output Value 2	Output Value 1	Output Value 0

When local logic is being used, DeviceLogix-based devices treat all consumed I/O data as “Network Inputs”. That is to say that the device may consume any amount of data, regardless of hardware, and treat it as generic input from the network. By default, the first N Network Inputs provide the data for the N hardware outputs found on the device. However, if you bind the hardware output to some other entity in the logic, the corresponding Network Input does not provide the control for that output.

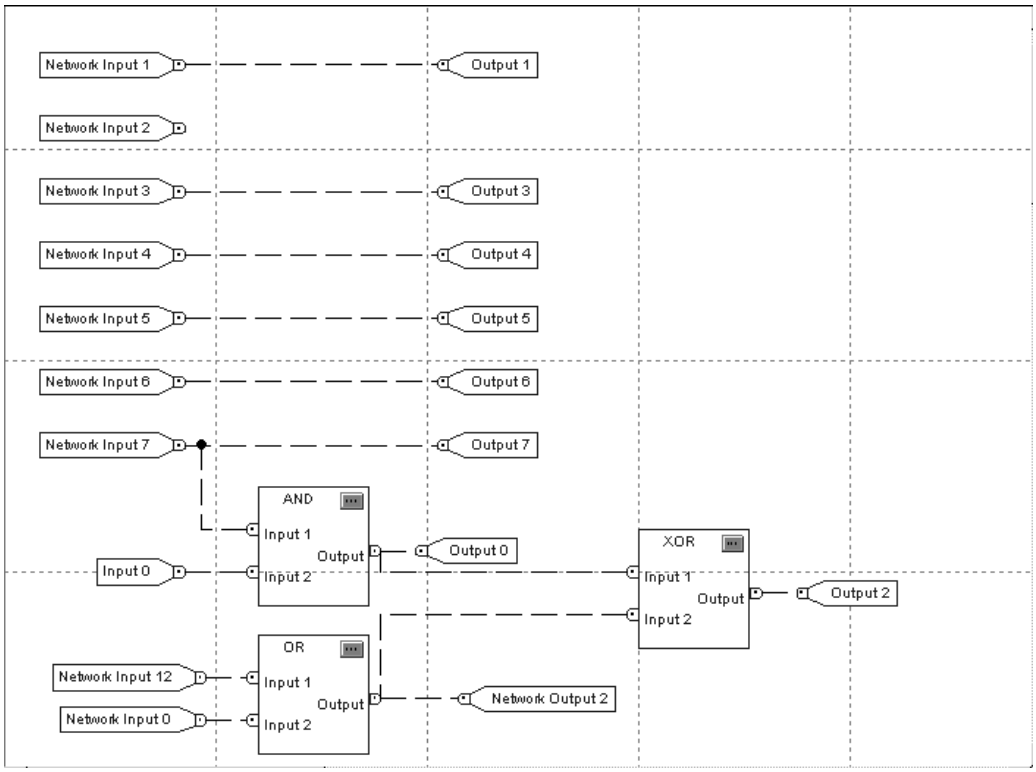
Each Network Input has a bit reserved for it in the consumed I/O assembly. The number of Network Inputs that a device consumes is not directly associated with the number of hardware resources present on the device. Therefore “extra” data can be sent to the module for use in the local logic. One such assembly is shown below for the same 8-output device considered above.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	Network Input 7	Network Input 6	Network Input 5	Network Input 4	Network Input 3	Network Input 2	Network Input 1	Network Input 0
Byte 1	Network Input 15	Network Input 14	Network Input 13	Network Input 12	Network Input 11	Network Input 10	Network Input 9	Network Input 8

Note that even though the device only contains 8 outputs, it is consuming 16 bits of information. You can then map additional information into the data that is sent to this device by the master controller.

The following list contains several important details about Network Inputs. These details are further summarized in the illustration that follows the list.

- If an output is not used in the local logic, there is an assumed connection to a corresponding bit in the consumed data. In the illustration below this connection is shown explicitly for outputs 1, 3, 4, 5, 6, and 7. Note this connection does not need to be made graphically with the configuration tool. The DeviceLogix based module assumes this connection until it is broken by making a connection to some other piece of information. If no connection is made to this output, it will continue to take its control from the master controller via its Network Input.
- Network Inputs can be used as input to more that one entity. The illustration below shows that the data sent to control output 7 is also used in the logic.
- When an output is bound to logic, its corresponding Network Input can be used somewhere else or not used at all. The illustration below shows Output 0 and 2 being controlled by logic. Output 0's corresponding Network Input (0) is being used as an input to the OR gate. Output 2's Network Input is unused.
- As mentioned earlier, the number of Network Inputs is not directly associated with the number of hardware outputs present on the device. The illustration below represents logic for an 8-output device. However, Network Input 12 is used in the logic. This bit may be the result of some logic in the master controller. It may also be a bit from another device on the network that has been forwarded to this device by the master via the DeviceNet scanner.



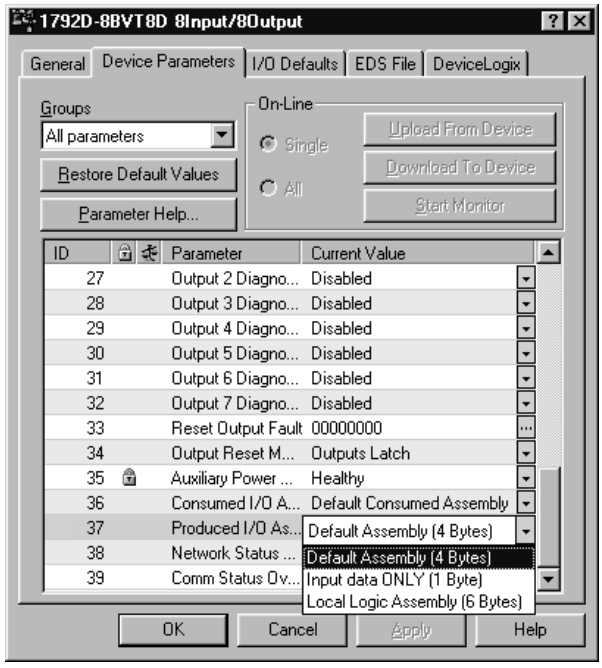
Network I/O extends the capabilities of I/O modules by allowing them to produce and consume the data needed in their application. Network I/O is critical to the effective use of DeviceLogix-based devices in a networked control system because of its ability to exchange information between the local control and external events being monitored by the master controller.

Determine Parameters

The following sections help you determine values for DeviceLogix parameters.

Determine Produced I/O Assembly

Like the consumed I/O assembly, the produced I/O assembly parameter lets you select how much data gets produced across DeviceNet. In order to have DeviceLogix specific data, produced by the device, select the entry **Local Logic Assembly**. As with the Consumed Data, you must make sure that the settings in the scanner's scanlist match your selection here. The example below illustrates this concept for an 8in/8out MaXum block.



Input 7	Input 6	Input 5	Input 4	Input 3	Input 2	Input 1	Input 0
OW-D	OW-C	OW-B	OW-A	ISC-D	ISC-C	ISC-B	ISC-A
OFLT 7	OFLT 6	OFLT 5	OFLT 4	OFLT 3	OFLT 2	OFLT 1	OFLT 0
	OPWR	Logic Ena					
OUT 7	OUT 6	OUT 5	OUT 4	OUT 3	OUT 2	OUT 1	OUT 0
PNB 7	PNB 6	PNB 5	PNB 4	PNB 3	PNB 2	PNB 1	PNB 0

Standard MaXum Status and Diagnostic Bits
OW = Off-Wire; ISC = Input Short Circuit; OFLT = Output Fault; OPWR = Output Power

DeviceLogix Bits
Logic Ena = DeviceLogix is Enabled. This can be used in the PLC to recognize that the remote device is running a control program.
Out X = Status of Local Output Bit if it is under control of DeviceLogix
PNB X = Network Output Bits
(Where X = the number of the Output Bit or Network Output Bit.)

To change a produced I/O assembly:

1. Click the current value of the Produced I/O Assembly.

   The field turns into a pull down list box.

2. Click the arrow to display the values.

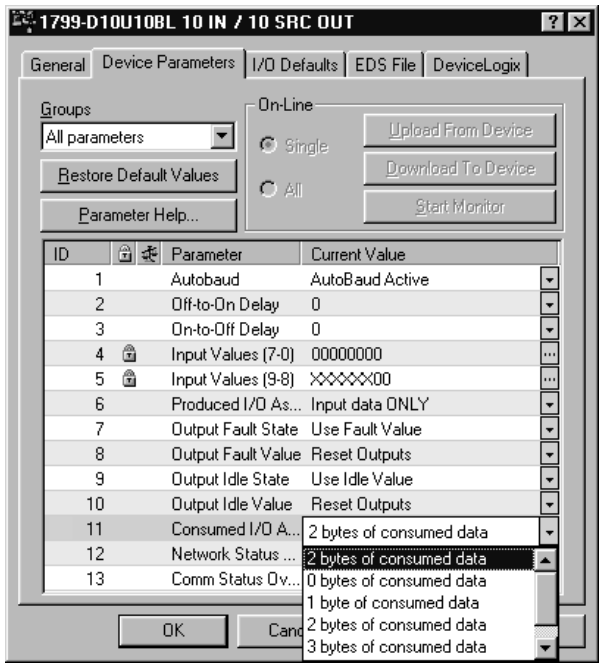
3. Click the desired value.

4. Click **Apply** to save the new value without closing the window.
   or
   Click **OK** to save the new value and to close the window.

For the layout of the Produced I/O Assembly for your chosen device, please refer to the Technical Data publication for that device.

Determine Consumed I/O Assembly

Devices, when enabled with DeviceLogix, can receive additional data across DeviceNet. This data is called Network Inputs. The Consumed I/O Parameter lets you select how much data you want the device to consume from a master. After setting this parameter, the same number of bytes must be selected in the scanner's scan list. For example, if you choose 3 bytes of data to be consumed, you must indicate that 3 bytes of data are to be transmitted in the scanner's scanlist. If the number of consumed bytes and the number of transmitted bytes do not match, the connection cannot be made. Please note that this parameter is not in all DeviceLogix enabled devices.



To change a consumed I/O assembly:

1. Click the current value of the Consumed I/O Assembly.

   The field turns into a pull down list box.

2. Click the arrow to display the values.

3. Click the desired value.

4. Click **Apply** to save the new value without closing the window.

   or

   Click **OK** to save the new value and to close the window.

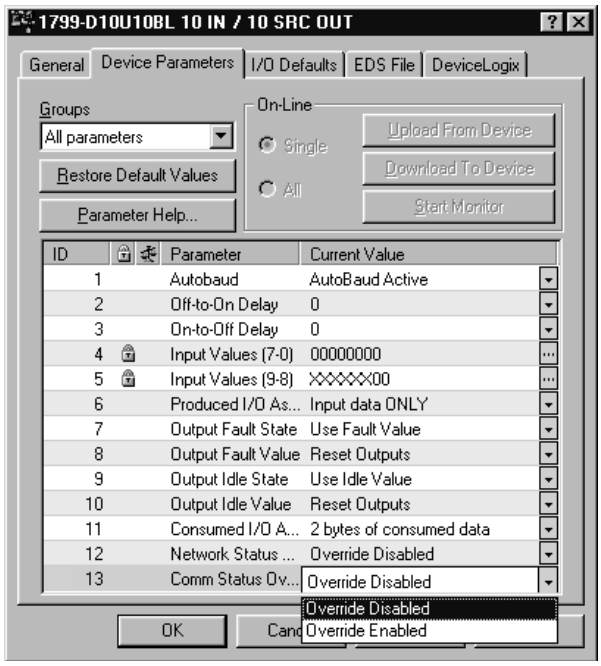
   To see the layout of the consumed I/O assembly for your chosen device, please refer to the Technical Data publication for that device.

Determine Comm Status Override

The Comm Status Override parameter controls whether local logic should control outputs when no “active” I/O connection exists with the device. You determine if the communication status override is enabled or disabled.

The Comm Status Override parameter is used to override normal behavior during the following events.

Event	Behavior with Comm Status Override Parameter Disabled	Behavior with Comm Status Override Parameter Enabled
Communications not established (module not on line). - OR - The module is on line but has no connections.	Output remains in the Available state until an I/O connection is established.	Local logic can update output values.
An I/O connection transitions to timed out state.	Output value is updated based on the output's Fault Action and Fault Value parameters.	Local logic can continue to update output values.
An I/O connection is deleted.	Output enters the Available state until a new I/O connection is established.	Local logic can continue to update output values.
An Idle is received.	Output value is updated based on the output's Idle Action and Idle Value parameters.	Local logic can continue to update output values.



IMPORTANT The above information is relative only to bound outputs.

To change comm status override:

1. Click the current value of the Comm Status Override parameter.

   The field turns into a pull down list box.

2. Click the arrow to display the values.

3. Click the desired value.

4. Click **Apply** to save the new value without closing the window.
   or
   Click **OK** to save the new value and to close the window.

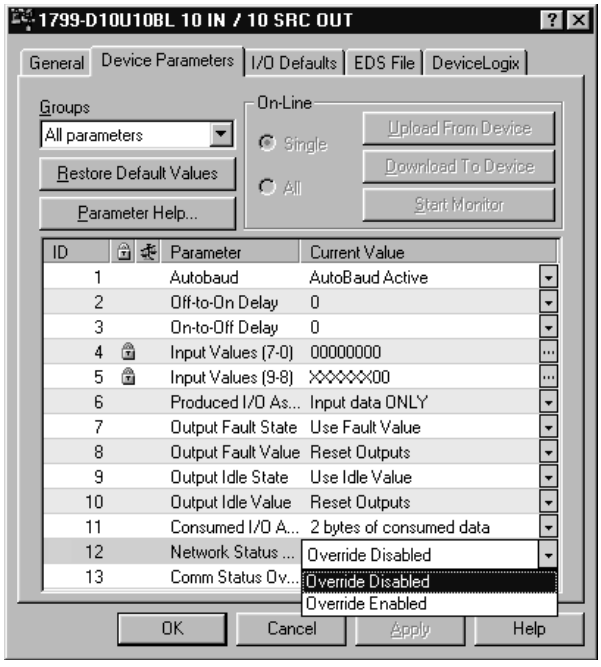
Determine Network Status Override

The Network status override parameter controls whether local logic should control outputs when it detects a duplicate machine ID or enters a bus off condition. You determine if the network status override is enabled or disabled.

When the override is disabled (default), the device will turn outputs off under any of the events. If the override is enabled, the device ignores any Network Fault. DeviceLogix stays in control of outputs if DeviceLogix is enabled, even if there is a Network Fault.

The Network Status Override parameter is used to override normal behavior during the following events.

Event	Behavior with Network Status Override Parameter Disabled	Behavior with Network Status Override Parameter Enabled
Duplicate MAC ID Failure	Module is put into an inoperable state and all outputs remain off.	Local logic can continue to update output values.
Entering the bus off state at power up	Module is put into an inoperable state and all outputs remain off.	Local logic can continue to update output values.
Entering the bus off state during run time	Module is put into an inoperable state and all outputs assume a safe state.	Local logic can continue to update output values.



To change network status override:

1. Click the current value of the Network Status Override parameter.

   The field turns into a pull down list box.

2. Click the arrow to display the values.

3. Click the desired value.

4. Click **Apply** to save the new value without closing the window.
   or
   Click **OK** to save the new value and to close the window.

Download To a device

When you change the size of a connection, you must download to the device for the change to take effect. When downloading to a device that a master is currently scanning you must:

1. Access the DeviceNet scanner's scanlist to remove the device from the scanlist.

2. Put the master in Idle mode. (This may mean turning a processor's keyswitch to Program.)

3. Access the device's parameters to change the size of the connection.

4. Download to the device.

5. Add the device to the scanlist.

6. Click the **Edit I/O Parameters** button and verify that the Rx and Tx size corresponds to the new size.

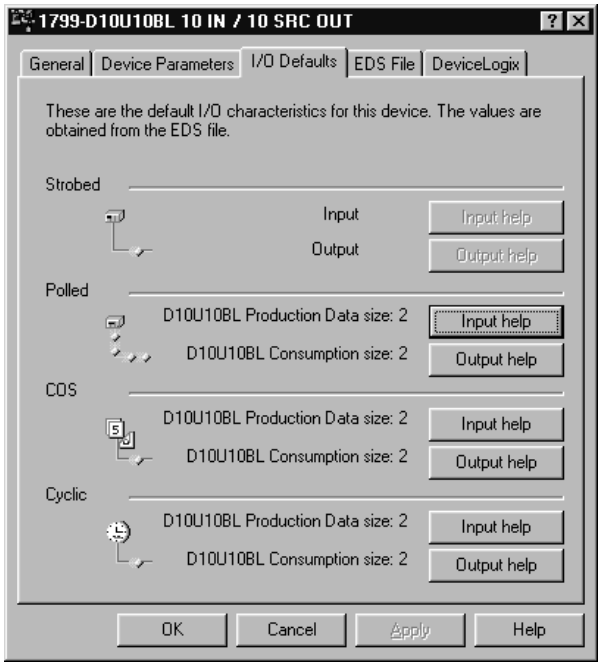
7. Put the master in Run mode for normal operation.

IMPORTANT If no response is received from the block, cycle power.

Access Default I/O Information

The **I/O Defaults** window provides information about default I/O characteristics for the device. The amount of data that the device reports is described here. Note however that the help only covers the data returned by the EDS file default settings. This window does not report data enabled by the DeviceLogix specific settings that were discussed above.

To access the **I/O Defaults** window, click on the **I/O Defaults** tab in the device properties dialog box.



To access I/O default information:

1. Click the **Input help** button to see input I/O default characteristics for the connection. Click the **Output help** button to see output I/O default characteristics for the connection.

   If the buttons are disabled for the connection, that connection is not available for the device.

2. To close the help window, click the **x** on the top right side of the window.

Access EDS Information

The **EDS File** window lists information about the EDS file for a selected device. Use this information to determine if the EDS file is the appropriate one for the device. Information such as Creation Time, Creation Date, Modification Time, Modification Date, and File revision are listed here. You can check this information against EDS files available from the web to determine if you have the most up to date version of the file. To access the **EDS File** window, click on the **EDS File** tab in the device properties dialog box.

Click the **View File** button to view the contents of the selected EDS file. The information stored in the EDS file is very detailed and is beyond the scope of this document. Although if you are having trouble enabling DeviceLogix features, you should check for the following line in the **[Device]** section of the EDS file:

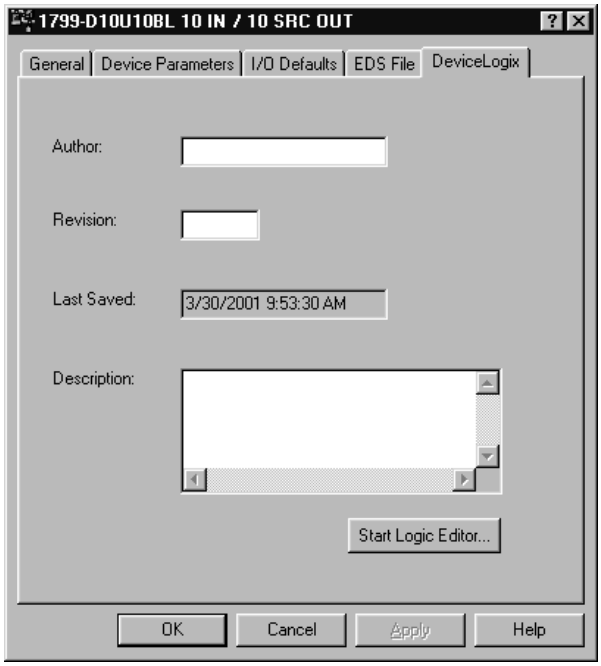
```
1_RSNetWorx_DNetTool = "LeoApp.LeoDevice";$Indicates LCT should be used on this device.
```

If this line is not part of the **[Device]** section of the EDS file for the device, the device will not be Devicelogix enabled.

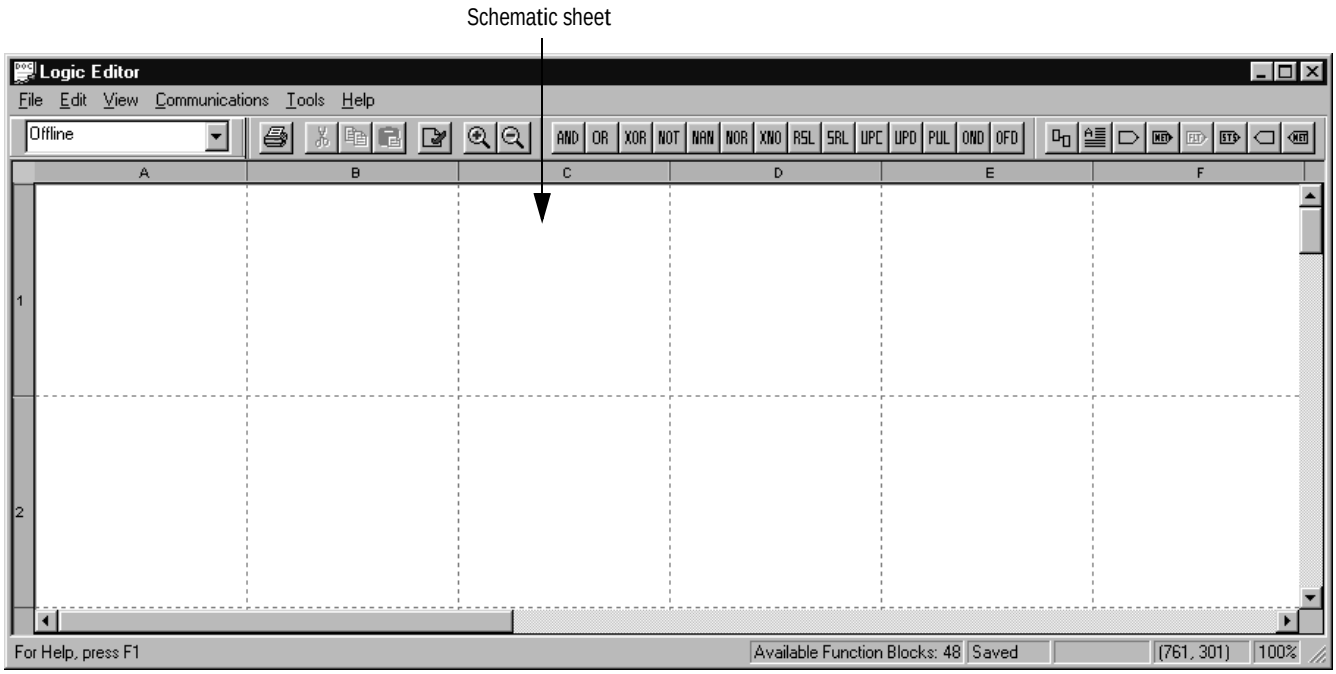
DeviceLogix EDS files will not work in a prior version than 3.0 of RSNetWorx for DeviceNet. DeviceLogix EDS files do not ship with devices or as part of the normal RSNetworx release. They are found on the CD that contains RSNetworx for DeviceNet 3.0 in a separate folder. Therefore, to enable DeviceLogix features for a DeviceLogix-capable device, you must register the EDS file for your particular device using the EDS wizard. See the section **EDS Files**, page 2-1, for more information.

Access DeviceLogix-Enabled Devices

Once you register the EDS file, you have access to the DeviceLogix Editor. You add DeviceLogix-enabled devices to your network just as you do for any other DeviceNet component. You see an additional tab in the device properties dialog box for all DeviceLogix-enabled devices. This new tab is labeled **DeviceLogix**. Click the **DeviceLogix** tab to access the start-up window for the Logic Editor. You can fill in your name, a revision number, and a description of your configuration (all optional fields). The **Last Saved** box updates every time you save within RSNetworx for DeviceNet.



Click the **Start Logic Editor** button in the start-up window for the Logic Editor to access the Logic Editor window.



Use DeviceLogix Editor Tools

The DeviceLogix editor lets you connect Function Blocks together to create the logic you want the device to perform. This section describes the tools designed to help you work easily within the DeviceLogix Editor while creating logic.

The tools available in the DeviceLogix Editor include:

- the schematic sheet
- menus
- toolbars

Only the features that are unique from standard Windows applications are discussed in detail. Functions that are common to all Windows applications, such as copy and paste, are not discussed.

Create Logic Using the Schematic Sheet

The schematic sheet is the area in which you place Function Blocks to create logic. The area is laid out on a grid with letters A through P representing the columns and numbers 1 through 16 representing the rows. This area is large enough to easily place all the Function Blocks, I/O tags, and associated connecting wires needed to create the desired logic.

TIP

The zoom level controls how much of the schematic you see at any given time. If you want to see more of the schematic, zoom out. If the you want more detail, zoom in. You can also use the scroll bars to move the schematic sheet around to display parts of the schematic that do not fit on the display.

The printed schematic uses 12 size A sheets of paper. The editor prints the schematic on four rows of three sheets. The schematic is printed at full size, regardless of the setting of the current zoom level. To make adjustments to the schematic before printing, you can use the print preview feature to see how the schematic will print. The schematic can also be sent to a plotter for easier viewing.

Use Menus Within the DeviceLogix Editor

Use menu options to create and maintain a schematic. The toolbar provides convenient shortcuts to the features available in the menu; however, all of the features of the schematic editor are available from the menu and not from the toolbar. See the **Display Toolbars** section, page 3-27, for more details about the toolbar. The main menu consists of the following items:

- File
- Edit
- View
- Communications
- Tools
- Help

The main menu items, and the submenu options available on each, are discussed below.

Use the File Menu

From the **File** menu, you can access print options and exit the DeviceLogix editor. You can **Print**, **Print Preview**, **Print Setup**, and **Close** from the **File** menu. The print, print preview, and print setup options provide the standard features of any Windows application and therefore are not discussed in detail. The close option closes the DeviceLogix Editor and returns you to the RSNetWorx for DeviceNet device properties dialog box.

Notice that there are no options to save, open, or create new logic schematic sheets on the **File** menu. The DeviceLogix Editor is not a standalone application. It is an add-on package to RSNetWorx for DeviceNet. As part of RSNetWorx for DeviceNet, it saves all of its information in the standard .DNT file; therefore all of the options that pertain to saving and opening new files are done in RSNetWorx for DeviceNet. There can be only one schematic per device in any RSNetWorx for DeviceNet network.

Use the Edit Menu

From the **Edit** menu, you can modify the schematic. The **Undo**, **Redo**, **Cut**, **Copy**, **Paste**, **Delete** and **Select All** functions adhere to the standards of all Windows applications. The unique features of the **Edit** menu are **Add Element**, **Properties**, and **Recovery Mode**. These unique elements are discussed in Chapter 5.

Use the View Menu

From the **View** menu, you can select the tools you wish to view as well as how much of the schematic you wish to see. The options available on the **View** menu are: **Toolbar**, **Status Bar**, **Zoom In**, **Zoom Out**, **Fit To Page** and **Zoom to 100%**. These options are discussed below.

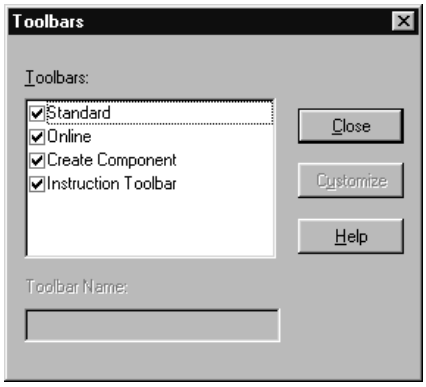
View Toolbars

Use the Toolbar option to control what toolbars display in the Logic Editor. The toolbar provides short cuts for accessing commands available in the menu. See the **Display Toolbars** section, page 3-27 later in this document for a description of the available options.

To modify the toolbar view:

1. Click **View>Toolbar**.

   The **Toolbars** window opens.



   Each toolbar by default is checked.

2. Click the box in front of the toolbar name to uncheck it.

   The toolbar disappears from Logic Editor.

3. Click **Close**.

   If you uncheck all of the toolbars, then the toolbar option is unchecked in the **View** menu.

   Note: You can also access the toolbar window by right clicking in the toolbar area.

Display the Status Bar

The status bar option controls the display of the status bar at the bottom of the application window. The status bar option is a simple on/off toggle. Click **View>Status Bar** to toggle the status bar on or off. When the status bar menu option has a check next to it, the editor displays the status bar at the bottom of the window. See the **Describe the Status Bar** section, page 3-33 for a more detailed description of the Status Bar.

Choose Zoom option

The following zoom options are available: **Zoom In**, **Zoom Out**, **Fit To Page**, and **Zoom to 100%**. Zoom controls how much of the schematic the application displays. Each zoom function is described below.

- **Zoom In** - enlarges a section of the drawing making it easier to see but causes less of the drawing to be visible.
- **Zoom Out** - displays more of the drawing but causes the elements of the drawing to appear smaller.
- **Fit to Page** - causes the entire logic diagram schematic to fit within the application window. This causes the logic elements to appear very small but displays the entire schematic.
- **Zoom to 100%** - causes the display of the elements on the screen to match the size of the elements on the printout.

Use the Communications Menu

The **Communications** menu contains online options only. The following options are available from the **Communications** menu: **Upload**, **Download**, **Logic Enable On**, and **Logic Enable Off**.

Upload Devices

Use the upload function to copy the configuration that is currently in the device into the editor. This causes any changes made on the screen to be discarded and for animation to occur. For a description of what happens during animation, please see **Edit On Line**, page 5-10. Click **Communications>Upload** to upload configurations.

Because a device has limited memory to store logic, the information about the x and y coordinates of Function Blocks is not stored in the configuration that is downloaded to the device. Therefore, the display of uploaded information will not match the appearance of information downloaded to a device. The position of the Function Blocks on the schematic sheet and information pertaining to the resolution of execution order in feedback loops are lost. However, the flow of the logic is correct.

When the editor uploads the schematic, it attempts to minimize the crossing of Function Block connections. Needless to say, the resulting layout of the schematic is radically different from the downloaded layout. For this reason, it is strongly recommended that, once you have your logic configuration complete, you print it for a record of how it looked when you created it.

The information pertaining to the resolution of the execution order is lost because the device only needs to know the execution order. It does not need to know how that order was determined.

For example, you create a schematic that includes a feedback loop. Then you resolve the execution order with an Assume Data Available flag on an input to a Function Block that is part of the loop. When you download the schematic to the device, the editor uses the Assume Data Available flag to determine the execution order of the Function Blocks. The editor then downloads the Function Blocks to the I/O device in the determined execution order. For more information on the Assume Data Available function, see page 5-7.

When you upload the logic, the device sends the Function Block information, including the execution order, but it does not have the flag to resolve future execution order issues. If you try to verify the application after the upload, the editor rejects it because it cannot resolve the feedback loop. You must again set the flag to tell the editor which block can assume that its data is available.

When you upload, the uploaded information is not automatically stored into the RSNetWorx for DeviceNet .DNT file. You must make sure you save any changes after you exit the DeviceLogix Editor to ensure that any changes made to the schematic are saved, including changes made as a result of an upload.

Download Devices

The Download option copies the configuration from the editor to the device. Click **Communications>Download** to download configurations. After the editor downloads the logic to the device, you have the choice to enable the logic or leave the logic disabled. When logic is enabled in the device, the download process occurs as follows:

- You select the download option from the **Communications** menu.
- The editor checks to see if the logic is enabled on the device.
- Because the logic is enabled, the editor prompts you if it is OK to disable the logic.
- If you answer YES, the editor begins to download the logic to the device.
- If you answer NO, the editor aborts the download and displays an error.

- Again, because the logic was enabled before the download, when the download completes, the editor asks if you wish to re-enable the logic.

  If logic was not enabled before the download, the editor does not prompt you to disable and re-enable the logic.
- When the download is complete, the editor displays a download successful message box.

  If the master is currently scanning the device to which you are downloading, you must:

1. Access the DeviceNet scanner's scanlist to remove the device from the scanlist.
2. Put the master in Idle mode. (This may mean turning a processor's keyswitch to Program.)
3. Access the device's parameters to change the size of the connection.
4. Download to the device.
5. Add the device to the scanlist.
6. Click the **Edit I/O Parameters** button and verify that the Rx and Tx size corresponds to the new size.
7. Put the master in Run mode for normal operation.

IMPORTANT If no response is received from the block, cycle power.

At any point during the download, you may click the **Cancel** button and the download aborts. If you click **Cancel**, the editor displays an error message and the device will contain no logic configuration. Neither the old application nor the new application exists in the device after you cancel during a download. You must perform another complete download in order to have the device execute logic.

Once the download is successful, the editor is able to animate the application. For a description of what happens during animation, please see **Edit On Line**, page 5-10. The actual state of all elements in the configuration are reflected by 1s or 0s displayed on the wires connecting the elements. If you left the logic disabled after the download, there will be no activity. However, the editor still displays the current state of the connections and the input can still change. If you enable the logic, the editor displays updated information every 500 milliseconds.

Even though the editor and the device have the same configuration after a download, remember that the information may not be saved in the RSNetWorx for DeviceNet .DNT file. To save the information in the RSNetWorx for DeviceNet .DNT file:

1. Exit the DeviceLogix Editor by selecting **File>Close** or clicking the **x** on the top right of the schematic.

2. Press **OK** in the device properties dialog box to close the box.

IMPORTANT You will lose all your logic if you press the **Cancel** button or the **x** on the top right of the box.

3. Click **File>Save** on the RSNetWorx for DeviceNet main menu.

   If you do not save changes in the RSNetWorx for DeviceNet .DNT file and download to the device, the device properties box requires you to upload or download the device the next time you try to switch to a different device properties window.

Use the Tools Menu

The **Tools** menu contains the **Logic verify** option. Use the **Logic verify** option to check for mistakes in a schematic's logic connections. Click **Tools>Logic verify** to verify your logic. **Logic Verify** checks if:

- all required Function Block inputs are connected
- an execution order can be determined

If any of the checks fails, the editor displays an error message. If the error is caused by a loop that prevents the editor from determining execution order, the editor highlights the errant loop connections in a different color. You should then use the Assume Data Available feature to tell the configuration in what order you want the blocks solved. See the **Assume Data Available** section, page 5-7. A detailed description of why the verify failed is shown in the **Message View** window of RSNetworx for DeviceNet. You must shrink the editor window to see the **Message View** window.

Use the Help Menu

The **Help** menu contains standard Windows help information about how to use the editor and about the editor itself. The help menu has the following options:

- **Help topics** - help specific to DeviceLogix in standard Windows format.
- **Function Block Help** - help specific to Function Blocks.
- **Release Notes** - information specific to the current release of the editor.
- **About DeviceLogix** - standard Windows About Box. The DeviceLogix About Box contains copyright information and two revision numbers. File revision identifies the revision of the DeviceLogix application DLL.

Display Toolbars Within the DeviceLogix Editor

The toolbars provide easier access to configuration options than menus. But if you have a low-resolution monitor, you may decide not to view toolbars and give more space to the schematic sheet. However, if you have a higher resolution monitor, you may decide to view toolbars and take advantage of the quicker access to the editor options.

You can also determine how and where the editor displays the toolbar. You can drag the toolbar to the top, bottom, or either side of the editor window. To drag a toolbar to a different location:

1. Click on the perimeter of the toolbar (anywhere except on a button).

2. While holding the mouse button down, move the mouse to place the toolbar.

   While you are dragging the toolbar around the application window, you see an outline of the toolbar. The outline changes shape to show how it will display in the current location. If the current location of the toolbar is along the perimeter of the editor window, the toolbar docks to the closest edge of the window. If the location of the toolbar is not near the edge of the editor window, the toolbar becomes a free-floating window. You can place toolbars in different locations. The editor remembers the location of the toolbars and places them in the same location each time the editor starts.

The toolbar saves space by only showing a small icon for each button. The icon represents the function the button performs. If you are not sure what a toolbar button does, move the cursor over the top of the button and the editor displays a popup text description of the button.

There are four toolbars:

- Standard Tools
- Instruction Toolbar
- Component Toolbar
- Online Toolbar

The names of the toolbars are not visible when the toolbars are docked; you see the name when the toolbars are free floating windows (placed anywhere within the schematic). The following sections describe the options available on the toolbars.

Standard Tools

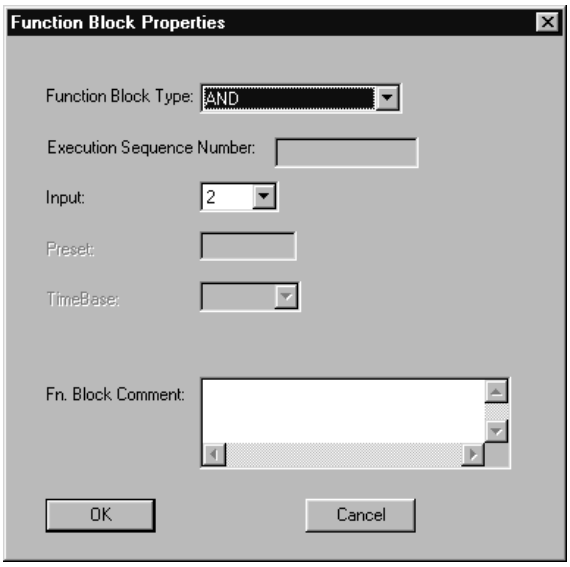
Most of the buttons available on the standard tools toolbar let you perform standard Windows type functions. The common tools include: **Print**, **Cut**, **Copy**, **Paste**, **Zoom In**, and **Zoom Out**. The only button on this toolbar that is not standard to Windows is the **Logic verify** button.

The **Logic verify** button performs the logic verify check on the schematic sheet. You can also perform this function by selecting **Tools>Logic verify**. For more information on the Logic verify option, See the **Use the Tools Menu** section, page 3-26.

Component

Use the component toolbar to add logic elements to the schematic. The options found on the Components toolbar are also found under **Edit>Add Element**. The options include:

Function Block - places a Function Block onto the schematic. A window displays containing the information that is needed to define the Function Block.



The fields in the **Function Block Properties** window are described below.

- **Function Block Type** - Use this field to select the type of Function Block to place onto the schematic. The drop down list to the right of the field contains all the types available for the device. The type of Function Block determines whether the **Input**, **Preset**, and **Time Base** fields are applicable.
- **Input** - This field is enabled only for the Boolean Function Blocks. Use this field to determine the number of inputs for the Boolean function. For example, an AND gate can have three inputs, in which case the output would be true only if all three inputs were true. The number of inputs supported for Boolean Function Blocks is product specific.

- **Preset** - This field is available for both counters and timers. For counters, the preset represents the number of "count up" events that must occur before the Function Block sets the output to TRUE. For timers, the preset represents the amount of time that the block must accumulate before it changes the output.
- **Time Base** - This field is only applicable for timer Function Blocks. Use this field to select the unit of time the preset represents. For example, if the Time Base is set to 1 millisecond, then a value of 10 for the preset represents 10 milliseconds. If the Time Base is set to 10 milliseconds, then the preset value of 10 represents 100 milliseconds.
- **Function Block Comment** - Use this field to add a text comment to the Function Block being created. The text comment appears directly above the Function Block when you place it onto the schematic.

For more information about the available Function Blocks, see **Create Logic With Function Blocks**, page 4-1. For details on how to configure Function Blocks, see **Create a Schematic**, page 5-1.

Text Box - places text onto the schematic. You can place text anywhere on the schematic.

To add a Text box:

1. Click on the text box button.
2. Move the cursor onto the schematic.

   The cursor changes from a pointer to a text entry cursor.
3. Click on the schematic where you want the upper left edge of the text to start.
4. Type your comment. You can type up to 256 characters.
5. Press ENTER to exit the text box.

Discrete Input Point - places a Hardware Input tag onto the schematic. The Hardware Input tag lets you connect Hardware Inputs to the inputs of Function Blocks. For a description of Hardware Inputs, see **About Hardware Inputs**, page 4-10.

Network Input Point - places Network Input tags onto the schematic. Network Inputs let you use data consumed from the network in your logic. For a description of Network Inputs, see **About Network Inputs**, page 4-10.

Fault Input Point - places fault input tags onto the schematic. Fault inputs let you modify the local logic in a device as a result of fault conditions. For a description of fault inputs, see **About Faults**, page 4-11.

Module Status Point - places Status Input tags onto the schematic. Module Status Inputs let you modify the local logic as a result of different status conditions in the module. For example, the local logic may behave differently if an explicit message connection exists between the device and another device. For a description of Status Inputs, see **About Status Inputs**, page 4-11.

Discrete Output Point - places a Hardware Output tag onto the schematic. The Hardware Output tag lets you connect Hardware Outputs to the output of Function Blocks. For details on Hardware Outputs, see **About Hardware Outputs**, page 4-11.

Network Output Point - places network output tags onto the schematic. Network outputs let you send information about the state of the local logic to a master device on the DeviceNet network. For details on network outputs, see **About Network Outputs**, page 4-11.

Instruction Toolbar

The instruction toolbar provides a shortcut to the available Function Blocks. For a description of each Function Block along with Truth Tables, see the **About DeviceLogix Components** section, page 4-1.

To select a Function Block from the Instruction toolbar:

1. Click the desired Function Block icon.

2. Click in the schematic to drop the Function Block.

Following is a description of each Function Block.

Function Block	Type of Function Block
AND	AND
OR	OR
XOR	exclusive OR
NOT	NOT
NAN	negative output AND
NOR	negative output OR
XNO	negative output exclusive OR
RSL	RS-Latch
SRL	SR-Latch
UPC	up counter
UPD	up and down counter
PUL	pulse timer
OND	on delay timer
OFD	off delay timer

IMPORTANT All of the Function Blocks may not be available for every product.

Online Toolbar

The options found on the Online toolbar are also found under **Edit>Communications**. You can perform uploads, downloads, logic enable on, and logic enable off functions. For more details, see the **Use the Communications Menu** section, page 3-23.

The online toolbar shows the current status of communications with the device and the status of the logic in the device. If you enter the editor while off line, you will see **Offline** in the toolbar. When you are working off line, none of the options on this toolbar are available. The options in the drop down list are grayed to indicate they are unavailable. When on line, the toolbar displays the current status of the logic in the device. When on line, the options in the drop down list are enabled. You will see **Logic Disabled** in the toolbar when the logic in the device is not running. You will see **Logic Enabled** in the toolbar when logic in the device is running.

Describe the Status Bar Within the DeviceLogix Editor

The status bar displays information about the status of the editor. The status bar option is a simple on/off toggle. To toggle the status bar:

1. Click **View>Status Bar** to toggle the status bar on or off.

   Or, click the right mouse button when the cursor is over the status bar.

   A menu displays with the toolbar and status bar options available.

2. Click the status bar option to toggle the status bar off.

   Note: You can only turn the status bar off by clicking the right mouse button over the status bar.To turn the status bar back on, click **View>Status bar**.

   The status bar contains six fields. These fields are described below, from left to right.

- The first field (left field) gives a brief description of the menu and toolbar items as you move the cursor.
- The second field shows the number of Function Blocks you may add to the schematic before you reach full capacity.
- The third field shows whether changes to the schematic have been saved to the RSNetWorx for DeviceNet .DNT file. If you didn't make edits to the schematic, this field displays "Saved". If you made changes since the last time you saved the file, the field displays "Not Saved".
- The fourth field shows whether the schematic in the editor matches the configuration in the device. If the two are different, the field displays "Pending Edit". If they are the same, the field displays "Animated" and the schematic reflects the live status of the running logic. If you are off line, the field is blank.
- The fifth field shows the x and y coordinates of the mouse cursor on the schematic.
- The six field shows the current zoom percentage. Normally the schematic is zoomed to 100%.

What's Next?

Now that you have learned how to access the RSNetWorx for DeviceNet device properties dialog box and the DeviceLogix Editor to set up general properties, determine parameters and use DeviceLogix Editor tools, read Chapter 4 to learn about the Function Blocks, inputs, and outputs that will aid you in creating logic.

DeviceLogix Components

What This Chapter Contains

This chapter contains information about Function Blocks, inputs, and outputs that will aid you in creating logic. The following table lists what this chapter contains and where to find specific information.

For information about	See page
DeviceLogix components	4-1
creating logic with Function Blocks	4-1
creating logic with inputs	4-10
creating logic with outputs	4-11

About DeviceLogix Components

DeviceLogix uses Function Blocks to let you configure the logic you want an I/O block to perform. The building blocks of the editor are the Function Blocks themselves and the inputs and outputs available to the device being configured.

Create Logic With Function Blocks

Function blocks are the building blocks used to create the logic that will reside in the device. The four functions that are available include:

- Boolean
- Bistable
- Counter
- Timer

IMPORTANT All of the Function Blocks are not available for every product.

Each of the blocks is described below. Before we discuss the individual Function Blocks, we must first describe how inputs and outputs work with Function Blocks.

Attach Inputs and Outputs To Function Blocks

The Function Blocks contain both inputs and outputs. The inputs to Function Blocks can be attached to any of the input types (see **Create Logic With Inputs**, page 4-10) or to the output of another Function Block. Only one connection can be made to a Function Block input and an input cannot be tied to another input.

Outputs of the Function Blocks can attach to either Hardware or Network Outputs (see **Create Logic With Outputs**, page 4-11) or to the inputs of other Function Blocks. A single Function Block output can be the source of (be connected to) any number of Hardware Outputs or Function Block inputs. Hardware Outputs cannot be tied to Hardware Outputs.

Configure With Function Blocks

The following sections describes how to configure the four types of function blocks. Truth tables further describe Function Blocks by listing all possible combinations of inputs and indicating the outputs for each combination.

Configure With Boolean Function Blocks

The Boolean Function Blocks are configurable based on the product. The Boolean functions that DeviceLogix supports include AND, OR, XOR, NOT, NAND, NOR, XNOR. Following is a truth table for a 2 input Boolean Function Block.

Truth table for the Boolean Functions

Function	Input1	Input2	Output Value
AND	0	0	0
	0	1	0
	1	0	0
	1	1	1
NAND	0	0	1
	0	1	1
	1	0	1
	1	1	0

Function	Input1	Input2	Output Value
OR	0	0	0
	0	1	1
	1	0	1
	1	1	1
NOR	0	0	1
	0	1	0
	1	0	0
	1	1	0
XOR	0	0	0
	0	1	1
	1	0	1
	1	1	0
XNOR	0	0	1
	0	1	0
	1	0	0
	1	1	1
NOT	0	N/A	1
	1	N/A	0

The NOT function can be accomplished by adding a Negate to a connection wire. This saves you from having to use a Function Block. For more information on the Negate function, see page 5-6.

IMPORTANT All Function Blocks require at least two input values with the exception of the NOT function which requires only one input value.

Configure With Bistable Function Blocks

The Bistable Function Blocks implement standard latches. These include the SR Latch (set dominant) and the RS Latch (reset dominant). Each latch block has two inputs called Input and Reset and an output called Output. Following is the truth table for the Bistable Function Blocks.

Truth Table for the Latch Functions

Function	Reset	Input	Output Value at time = t_0	Output Value at time = t_{0+1}
SR (set dominant)	0	0	0	0
	0	1	0	1
	1	0	0	0
	1	1	0	1
	0	0	1	1
	0	1	1	1
	1	0	1	0
	1	1	1	1
RS (reset dominant)	0	0	0	0
	0	1	0	1
	1	0	0	0
	1	1	0	0
	0	0	1	1
	0	1	1	1
	1	0	1	0
	1	1	1	0

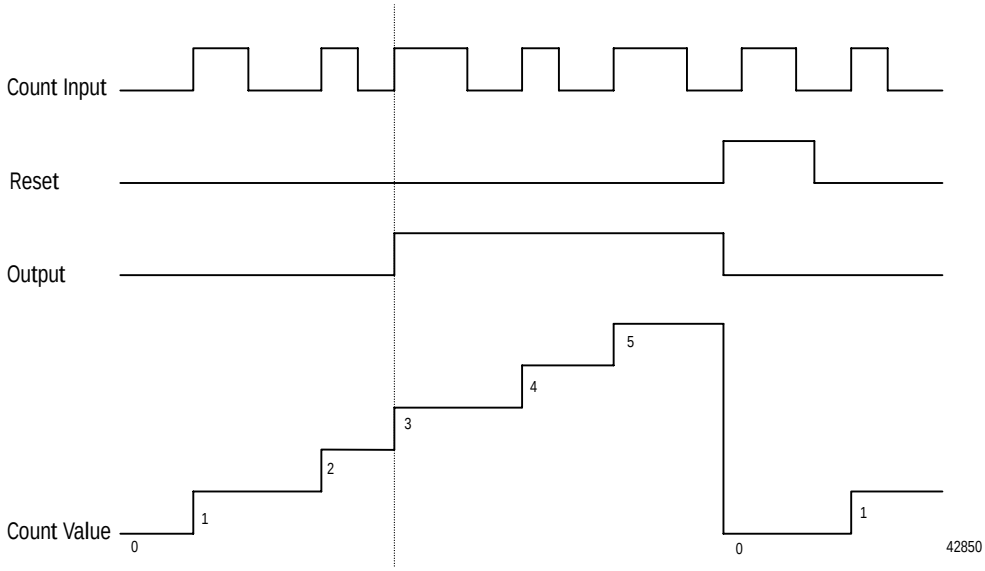
Configure With Counter Function Blocks

Counter Function Blocks count events that happen inside the local logic of the I/O block. DeviceLogix supports Up counters and an Up/Down counter. Both the counters have a preset that can be set to between 0 and 65,535. They both have accumulators that store the current event count. The counters are non-retentive. When the count reaches 65,535, it stops until it is reset.

Configure With the Up Counter

The Up Counter has two inputs called Input and Reset and one output called Output. The Up counter simply counts up on a FALSE to TRUE transition and sets its output to TRUE when the Accumulator reaches the Preset value. The Accumulator continues to count up until the counter is reset or it reaches 65,535. A TRUE value on the reset input zeros the accumulator and sets the output to FALSE. An illustration of the counter function is below.

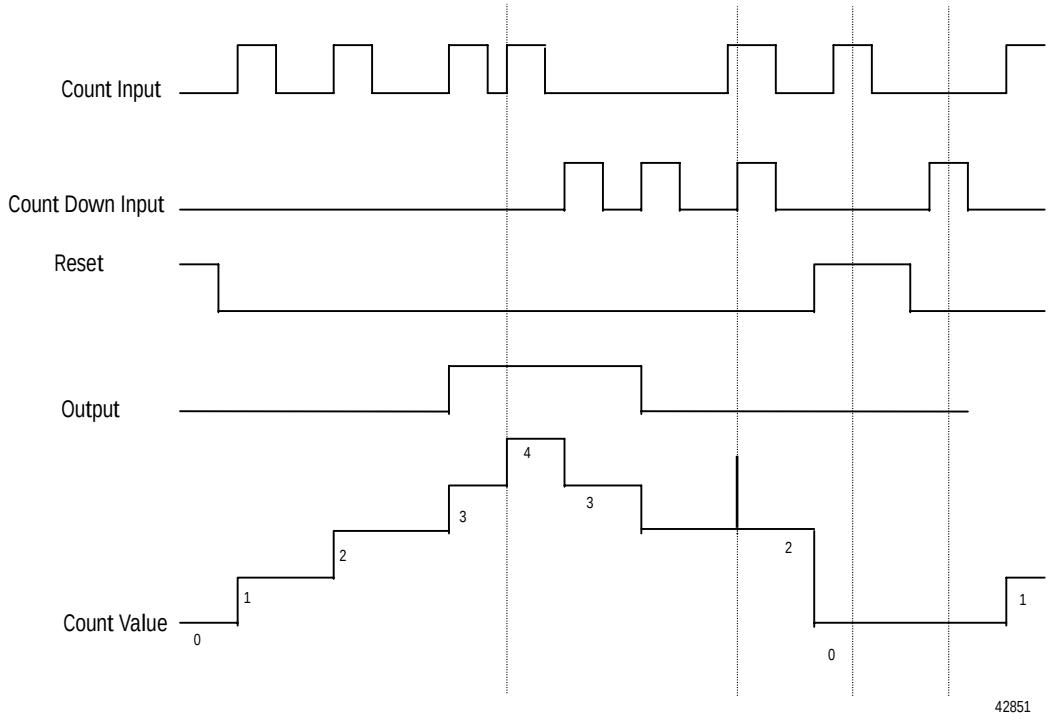
Timing Diagram for the Counter Function with a Preset Value=3



Configure With the Up/Down Counter

The Up/Down counter has three inputs called Input, Reset, and Count Down Input and one output called Output. The counter increments the accumulator any time the Input changes from FALSE to TRUE and decrements the counter any time the Count Down Input changes from FALSE to TRUE. When the accumulator is above or equal to the Preset value, the counter sets its Output to TRUE. When the accumulator falls below the Preset, the counter resets its Output to FALSE. Like the Up counter, the Up/Down counter resets when a TRUE level is detected on the Reset input. When a reset occurs, the counter zeros the accumulator and sets the Output to FALSE.

Timing diagram for the Up/Down Counter with a Preset Value=3



Configure With Timer Function Blocks

The Timer Function Blocks are similar to the Counter Function Blocks in that they control the state of a single output based on the value of an Accumulator and a Preset. They also have two inputs called Input and Reset which control the counting of the Accumulator. Unlike the counters, the timers accumulate time instead of events. The three timer blocks available are the Pulse Timer, the On-Delay Timer and the Off-Delay Timer. All three timers have a selectable timebase of either 1 millisecond or 10 milliseconds. The timers are non-retentive. Only the output status is available. No Timer timing or Timer Done Bits are available. The different timers are described below.

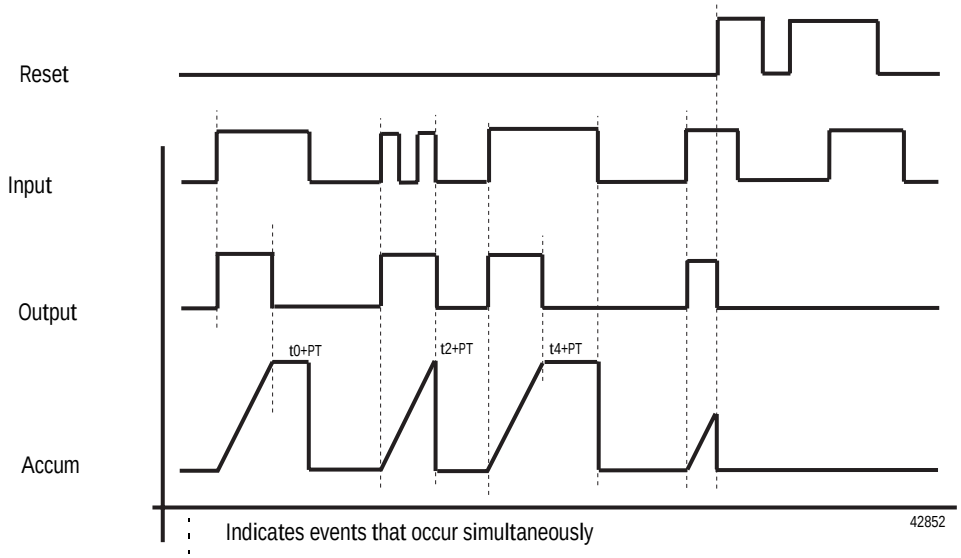
Configure With the Pulse Timer

The Pulse Timer generates a TRUE value on its output for a fixed amount of time. The duration of the TRUE pulse is determined by the Preset value along with the timebase of the timer. When the Input to the timer changes from FALSE to TRUE, it sets the Output to TRUE and starts the Accumulator counting. It then increments the Accumulator each time the timebase number of milliseconds has expired. When the Accumulator reaches the Preset value the Timer resets the Output to FALSE.

In the Pulse Timer, the Input acts only as a trigger to start the Accumulator counting. Once the Accumulator starts timing, it continues to rise regardless of the state of the Input. As long as the Output is TRUE and the Accumulator is counting, additional triggers of the Input do not affect the state of the Output or the count of the Accumulator. Once the Accumulator reaches the Preset value and the Timer resets the Output to FALSE, the Pulse Timer can again trigger the process by sensing a FALSE to TRUE transition on the Input pin. Even if the Input remains on the entire time the Accumulator is counting, when the Accumulator reaches the Preset value the Timer resets the Output to FALSE.

At any point during the operation of the Timer, if it detects a TRUE level on the Reset input it will disable the Timer and set the Output to FALSE. The Timer must again be triggered by a FALSE to TRUE transition on the Input. This means that if the Input is TRUE and the Timer is reset, the Timer will remain inactive. Even if the Reset changes back to FALSE while the Input is TRUE, the Timer remains inactive. In order to start a new pulse operation, the Input must change to FALSE and then back to TRUE.

Timing diagram for the Pulse Timer

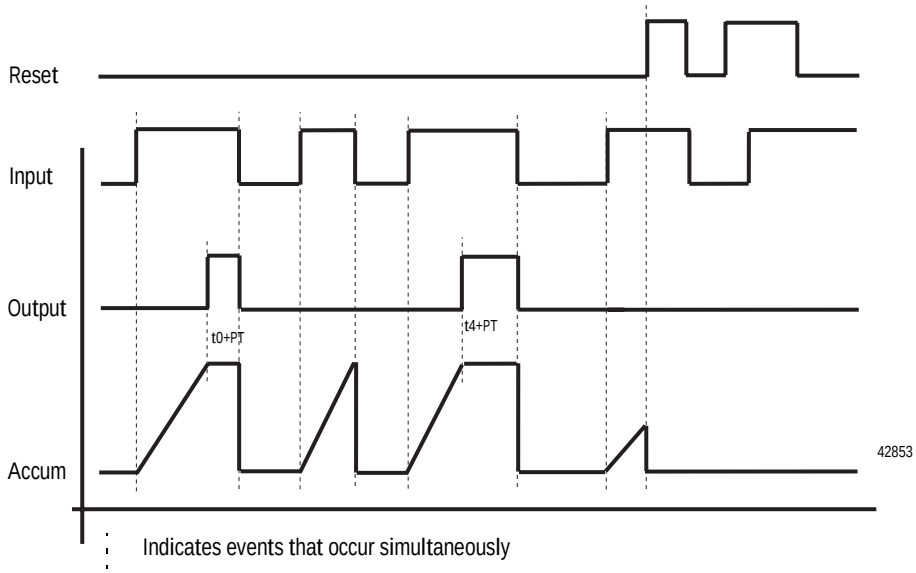


Configure With the On Delay Timer

The On Delay Timer delays the output response to an input by a desired amount of time. When the Input is TRUE, the Timer increments the Accumulator and when the Input goes FALSE the Timer resets the Accumulator. Each timebase unit of time, the Timer increments the Accumulator. When the Accumulator reaches the Preset value, the Timer sets the Output to TRUE. The Timer maintains the Output TRUE status as long as the Input remains TRUE. When the Input changes from TRUE to FALSE, the Timer resets both the Output and the Accumulator. If the Input goes FALSE before the time period specified by the timebase and Preset, the Output remains FALSE and the Accumulator is cleared, essentially ignoring the Input.

If the Timer senses a TRUE level on the Reset input at any time during the operation of the Timer, it resets the Output to FALSE and clears the Accumulator. Because the reset line is level sensitive, the timer remains reset until the Timer detects a FALSE on the reset input. Also, because the Input is level sensitive, the Timer will again begin to increment the Accumulator if the Reset line goes FALSE while the Input remains TRUE.

Timing Diagram for the On-Delay Timer

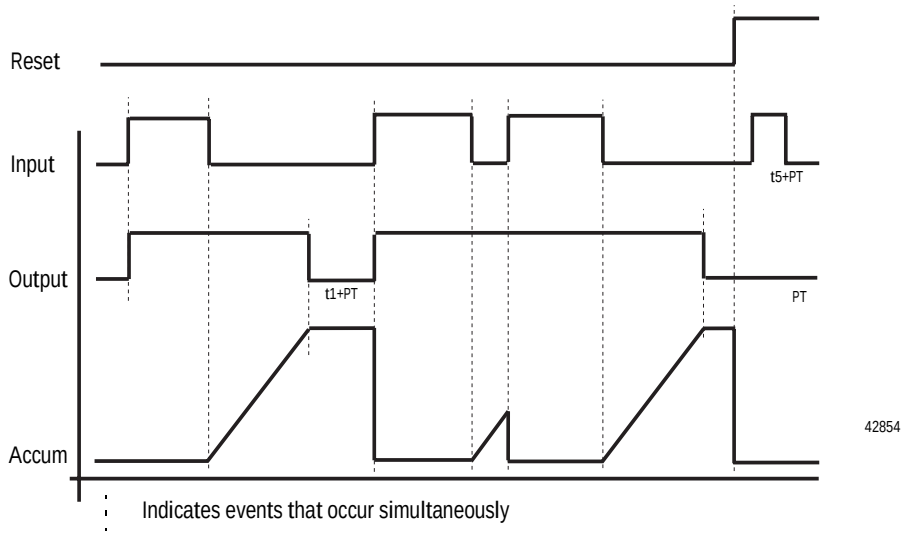


Configure With the Off Delay Timer

The Off-Delay Timer works the same way as the On-Delay Timer but instead of delaying the TRUE status of the Output, it delays the FALSE status of the Output. The Input to this Timer is a level sensitive FALSE with an edge-triggered reset on the FALSE to TRUE transition. This means that when the Input is FALSE, the Timer increments the Accumulator and when the Input goes TRUE the Timer resets the Accumulator. Each timebase unit of time, the Timer increments the Accumulator. When the Accumulator reaches the Preset value, the Timer sets the Output to FALSE. The Timer maintains the output FALSE status as long as the Input remains FALSE. When the Input changes from FALSE to TRUE, the Timer sets the Output TRUE and resets the Accumulator. If the Input goes TRUE before the time period specified by the timebase and Preset, the Output remains TRUE, and the Accumulator is cleared, essentially ignoring the Input.

If the Timer senses a TRUE level on the Reset Input at any time during the operation of the Timer, it resets the Output to FALSE and clears the Accumulator. Because the reset line is level sensitive, the Timer remains reset until the Timer detects a FALSE on the Reset input. Also, because the Input is level sensitive, the Timer will again begin to increment the Accumulator if the reset line goes FALSE while the Input remains FALSE. However, because the reset logic already set the Output to FALSE, the time delay causes no effect because the Output is already FALSE. If the Input is TRUE when the reset goes FALSE, the Timer sets the Output to TRUE.

Timing Diagram for the Off-Delay Timer



Create Logic With Inputs

Device inputs can connect to any Function Block input defined above or can drive an output directly. The number and type of inputs varies from device to device. However, all devices may support the following four categories of inputs:

- Hardware
- Network
- Status
- Fault

Each of these Inputs is described below.

About Hardware Inputs

Hardware inputs represent the actual inputs such as sensors and switches attached to a particular device.

About Network Inputs

Network inputs are data bits sent from a master that can be used in the device's logic.

About Status Inputs

Status inputs indicate the state of the device. For example, if an explicit message connection exists between the device and a master, a "explicit connection exists" input would be set to true and could affect the logic the device performs.

About Faults

Faults are conditions that report device errors. For example, if a device detects a short circuit on an output, it can set a fault input to true. The fault input can then affect the logic the device performs.

Create Logic With Outputs

DeviceLogix device outputs can connect to the output of any of the Function Blocks defined above or can be driven directly by an input. There are two types of outputs:

- Hardware
- Network

About Hardware Outputs

Hardware outputs are the actual outputs, such as lights and actuators, attached to a particular device. Without DeviceLogix, the master would normally control the outputs via consumed data. In fact, if there is no local logic controlling an output, the master controls the output as it would if DeviceLogix were not running on the device. However, within DeviceLogix, if the local logic controls an output, the master no longer controls the output. The only way the master can affect the state of an output that is under local control is to route requests to the local logic by using network inputs. Some outputs can be under local control while others can still be controlled by the master.

About Network Outputs

Network outputs report the results of the local logic to a master. These outputs can be attached to any output point on any Function Block and are part of the produced data from the device. They can also be connected to status, faults, or inputs.

What's Next?

Now that you have learned about the Function Blocks, inputs, and outputs that will aid you in creating logic, read Chapter 5 to create logic using the DeviceLogix Editor.

Create Logic Using the DeviceLogix Editor

What This Chapter Contains

Read this chapter to learn how to create logic using the DeviceLogix Editor.

To learn how to	See page
create a schematic	5-1
determine the state of the connection	5-6
interface a DeviceLogix Device to a Scanner	5-8
interface a DeviceLogix Device to a PC	5-10
edit on line	5-10
troubleshoot and debug	5-12

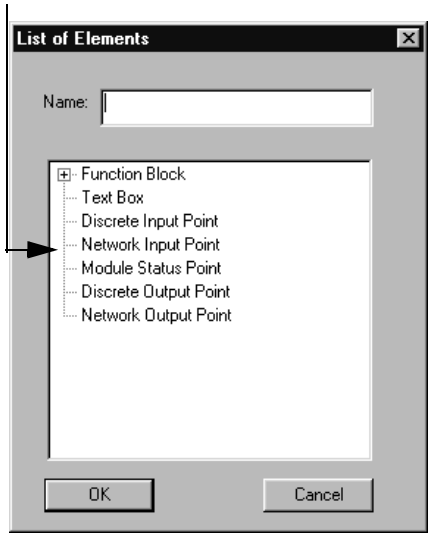
Create a Schematic

Add Elements To the Schematic

As described in chapter 3, you can add elements (Inputs, Outputs, Function Blocks, Text Boxes) to the schematic using the Toolbar or the menus.

Add an Element

List box



Use the **Add Element** option to add a new Function Block, input, or output to the schematic. To add a new element:

1. Select **Edit>Add Element**.

   You see the **List of Elements** box.

   The dialog box opens with the cursor already in the **Name** field.

2. Type the name of the desired Function Block or element in the **Name** field or click on the Function Block, or other element, in the list box below the **Name** field.

The list of available elements appear in the list box. To select a Function Block:

1. Click on the plus symbol next to the name **Function Block** in the list box.

   The list expands to include the names of all the available Function Blocks.

2. Click on the desired Function Block.

   The Function Block is added to the **Name** field.

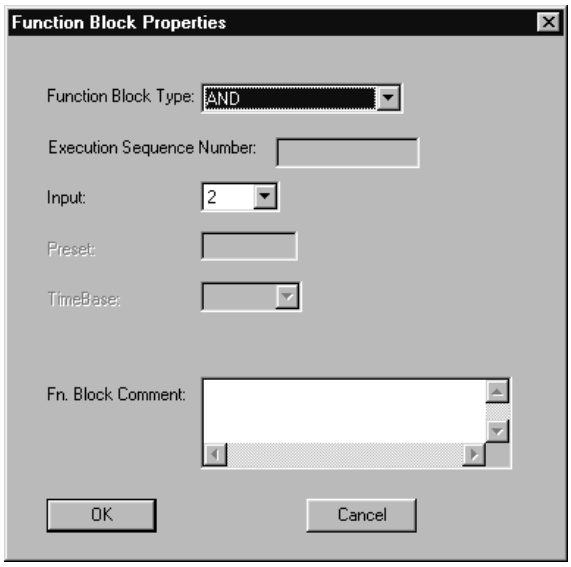
3. Click **OK**.

4. Click in the Logic Editor schematic to place the Function Block.

Another way to select a Function Block:

1. Double-click on the name **Function Block** in the list box.

   The **Function Block Properties** window opens.



2. Use the arrow on the right of the **Function Block Type** field to select a Function Block.

3. Enter any other applicable information in the remaining fields.

4. Click **OK**.

5. Click in the Logic Editor schematic to place the Function Block.

Select an Element Other Than a Function Block

To select an element other than a Function Block:

1. Click on the element in the **List of Elements** window.

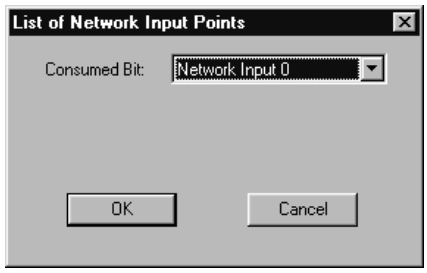
   The element is added to the **Name** field.

2. Click **OK**.

Another way to select an element:

1. Double-click on the name of the element in the list box.

   The elements properties window opens. The Network Input Point is used here.



2. Use the arrow on the right of the element's name to select a value.

   This box displays values for points that have not already been placed and lets you select the appropriate one for your selection.

3. Click **OK**.

4. Click in the Logic Editor to place the element.

   For more details about an element, please see the section **About DeviceLogix Components**, page 4-1. Because the values for each element vary based on the device being configured, this manual does not discuss the values for each element. Please refer to the Technical Data publication for the selected I/O device for information about these options.

   When you place the Function Block or other element in the Logic Editor, the cursor changes. If the element is a text box, the cursor changes to a text entry cursor. If it is any other element, the cursor becomes an outline of a Function Block.

Place an Element Into the Schematic

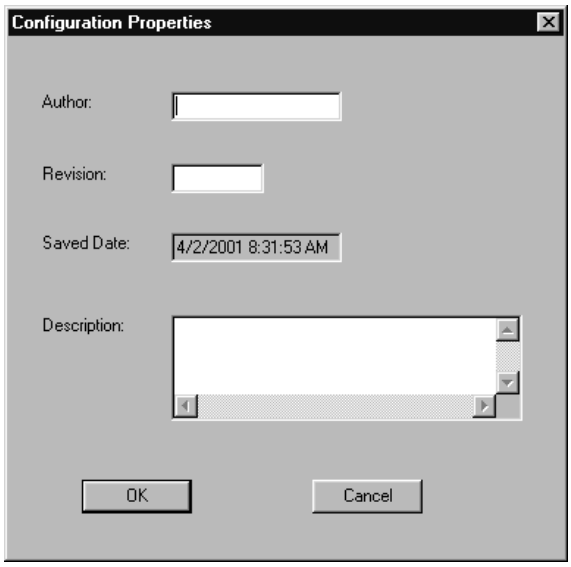
To place the element onto the schematic:

1. Move the cursor to the desired position on the schematic.
2. Click the mouse button once.

   The upper left position of the cursor identifies where the element will be placed. If you wish to remove the element once it is placed in the Logic Editor, press the ESC key. (Make sure the element is selected.) The element is removed and the cursor turns back into an arrow.

Modify Properties

Choose **Edit>Properties** to modify the **Author Name**, **Revision** or configuration **Description**.



Select Recovery Mode

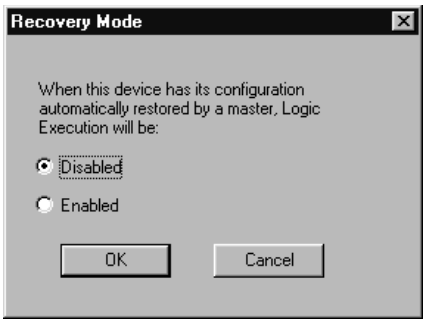
When using the Allen-Bradley master, choose the recovery mode option to determine how the I/O device should recover from an automatic download by a master device.

When an I/O device fails and a new device is added to replace it, a master device on the network can automatically download the stored configuration and logic to the device. This feature is known as Auto Device Replace (ADR). The recovery mode determines whether or not to enable the local logic when a download of this type occurs. The recovery mode is disabled by default. ADR in the scanner is also disabled by default.

To determine the recovery mode:

1. Choose **Edit>Recovery Mode**.

   You see the **Recovery Mode** window.



2. Click **Enabled** to enable local logic after a recovery. ADR must still be enabled in the master's configuration. Click **Disabled** and the automatic download will not occur.

3. Click **OK**.

Make Connections

Once you have Function Blocks and I/O points represented on the screen, you want to connect them to actually create the flow of the logic. The following section describes how those connections are made. To connect points:

1. Place the cursor on a connection pin (either on an I/O point or a Function Block) to cause the arrow to change to the connection cursor.

   The connection pin turns green if it is a valid spot on which to make a connection.

2. Click on the pin.

3. Move the cursor to the desired ending point for the connection.

   If it is a valid connection, that pin will turn green.

4. Click on the connection pin.

   The line (wire) will turn green indicating a good connection.

5. Click elsewhere on the grid and the wire will turn into a dashed line, indicating a good connector.

   When making connections, keep the following rules in mind:

- You can attach a Function Block output pin to any Function Block input pin. This gives you feedback capability.
- You can tie a Hardware or Network Input pin directly to a Hardware or Network Output pin.
- You cannot tie a Hardware or Network Output connection pin to a Function Block input pin.
- You cannot attach a Function Block output pin to any Hardware or Network Input pin.

Delete a Wire:

1. Click on the wire (it will turn green).
2. Click the Cut icon or press the DELETE key on your keyboard.

Determine the State Of a Condition

Once a connection is made, the Negate and Assume Data Available options effect how the state of that condition is determined.

Negate Data

The Negate function causes the status of the data to be negated before it enters the Function Block:

To negate data:

1. Right-click on the wire.

   You see a pop-up menu.

2. Click on **Negate**.

   The Negate menu option will have a check to the left of it. A circle will be placed on the wire next to the Function Block.

To cancel the Negate:

1. Right-click on the wire again.

   You see the pop-up menu.

2. Click on **Negate**.

   The check disappears from in front of the Negate menu option. The circle disappears from the wire.

   Note: Using the Negate option does not cause you to use one of the available Function Blocks, as would using a Not Function Block.

Set Assume Data Available

The Assume Data Available option assists the editor in determining Function Block process order when feedback paths are involved. When feedback paths are used, the Logic Editor's execution order algorithm may be unable to determine which Function Block needs to be resolved first. In this case, you must select which block will be evaluated first by designating that the editor should "assume that data is available" at that input. Once selected in this way, a double-headed arrow appears at the end of the wire indicating that it has precedence in the logic. You may be told by the system after a Verify that it could not resolve the Execution Order. (For an explanation of Logic Verify, see the **Use the Tools Menu** section, page 3-26.) In that case you must use the Assume Data Available Feature to resolve the conflict.

To set the Assume Data Available feature on a function block:

1. Right-click on the wire.

   You see a pop-up menu.

2. Click on **Assume Data Available**.

   The Assume Data Available menu option will have a check to the left of it. A double-headed arrow will be placed on the wire next to the Function Block.

To cancel the Assume Data Available:

1. Right-click on the wire again.

   You see the pop-up menu.

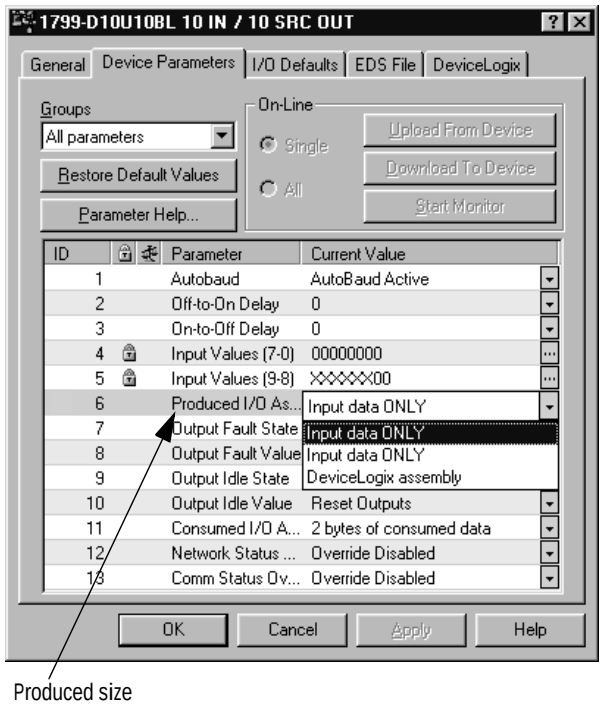
2. Click on **Assume Data Available**.

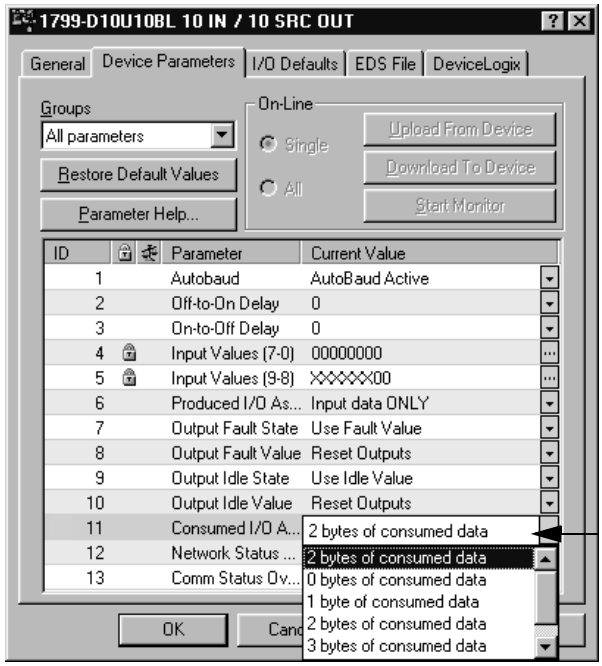
   The check disappears from in front of the Assume Data Available menu option. The double-headed disappears from the wire.

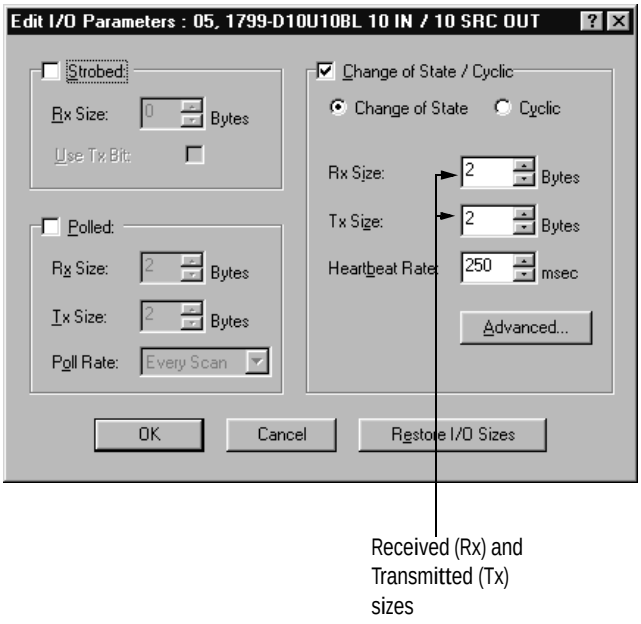
Interface a DeviceLogix Device To a Scanner

In order to communicate over DeviceNet, a DeviceLogix module must communicate through a DeviceNet scanner. The data from the DeviceLogix module must be mapped to an area in the scanner's Data Table. In the Device Parameters and I/O Defaults sections, we discussed how to set the amount of data the module is to produce and consume over DeviceNet (see pages 3-4 and 3-16, respectively). When the module is mapped into the scanner, care must be taken to assure that the Received (Rx) and Transmitted (Tx) sizes in the scanner are the same as the Produced and Consumed sizes specified in the Parameters section for the device. If you fail to do this, no data will be communicated. Consult the Users Guide for the scanner you are connected to in order to learn more about mapping modules correctly.

The following screens illustrate how to interface with a scanner.







Interface to a DeviceLogix Device Without a Network

Once configured, products containing DeviceLogix can operate without a network to perform small local control functions. It is advisable to continue to use standard DeviceNet cabling in these applications to accommodate the device's power requirements and future configuration needs.

If a Network is not used, ENABLE the Comm Status Override Parameter on the Device Parameters Tab. Doing this overrides the errors that would generate because there is no Network. If this parameter is not enabled, the Outputs will not react to the local logic. Note that in this setup, Network Inputs and Network Outputs are meaningless because there is no scanner in which to communicate.

Edit On Line

Once you've created your logic and set up your Scanlist or Override Parameters, you are ready to go on line.

To go on line:

1. Click the **On line** button on the RSNetworx toolbar.
2. Once the network browse is completed, double-click the device with which you wish to interact.
3. Click the **DeviceLogix** tab in the device properties dialog box.

   You may be asked to Upload or Download if you haven't already done so.
4. Once the upload or download is completed, click the **Start Logic Editor** button.

   You see the local logic resident in the device. You see the real-time status of the inputs and outputs indicated by 0s and 1s on the connecting wires. If you have Timers and/or Counters, you also see the value of the Accumulators change based on the logic. The On line Toolbar shows either Logic Enabled On or Logic Enabled Off, depending on the state of the device. You can change the state by simply selecting the desired state. From that Toolbar, you can also upload or download logic. In the Status Bar at the lower part of the screen, you see Animated, which means the logic on the screen matches the logic in the device. If you do not see Animated and you want the logic on the screen to match the logic in the device, perform an upload or download to synchronize the logic.

Edit While On Line

While the system is on line, you can make changes to the Timer and Counter values and to the program itself.

Change Timer and Counter Values

You can change the value of the Timer and Counter, Presets and Accumulators while the logic is running.

To change the value:

1. Double-click on the face of the Function Block you want to change.
2. Enter the new value in the box.
3. Press the ENTER key.

   The new value takes immediate effect. If you are changing a preset, remember to Save the configuration the next time you exit from the editor.

Change Logic

While the editor is running and animated you can change the logic that appears on the schematic. As soon as you do this, the editor will change from Animated mode into the Pending Edits mode. The logic in the device will not be affected and the current program will continue to run. However, the screen will not reflect the real-time status of the logic. The data on the wires will be frozen to whatever it was when you modified the logic. To ensure your changes take affect:

1. Select **Download** from the **Online** toolbar or select **Communications>Download**.

   You will be told that Logic is running and you will be asked if you want to stop it.

2. Select **No** and the download will not occur. The logic will continue to run in the device.
   Select **Yes** and the logic will stop running. The download will occur. When it is complete, you will be asked if you wish to start the logic running again.

3. Select **No** and the On line toolbar will show Logic Enable Off. The device will not run.
   Select **Yes** and the new downloaded logic will begin to operate. The On line toolbar will show Logic Enabled On. The schematic will again be animated. Not Saved will display in the status bar until you exit the editor and save the new configuration within RSNetWorx for DeviceNet.

Set Logic Enable On

Use the **Logic Enable On** function to tell the device to start processing the stored logic diagram. When the logic is disabled, the I/O device acts as if it is in an idle state. This turns on the Logic Enable Bit in the produced data. Once connections are complete, you can verify the logic. For a description of the Logic Verify function, See page 3-26. An implicit verify is performed on the logic when logic is enabled.

To set Logic Enable On, select **Communications>Logic Enable On**.

Set Logic Enable Off

Use the **Logic Enable Off** option to tell the device to stop executing the logic diagram. The device enters an idle state when you select this option. This turns off the Logic Enable Bit in the produced data.

The Logic Enable bit in the Produced I/O assembly of the device reflects a 0 or 1 for Logic Enable Off or Logic Enable On.

To set Logic Enable Off, select **Communications>Logic Enable Off**.

Troubleshoot and Debug

Forcing

To aid in troubleshooting and debugging of your schematic, you can Force Hardware Inputs and Hardware Outputs. No other inputs or outputs can be forced within the editor. Network Inputs and outputs can be forced in the device from where they originate. If you exit the editor with a force enabled, that force will stay in effect until you remove it.

To apply forces:

- Right-click on the input or output Element you wish to force.

  A pop-up appears that lists the forcing options, as described in the following sections.

Force Inputs

The following list describes the Input force options.

- Force On - forces the input ON.
- Force Off - forces the input OFF.
- Remove Force - return control of the input to the hardware device and return the element back to white.
- When forced, the Input Element turns yellow and the status value on the wire reflects the Force state chosen.

Force Outputs

Forcing outputs works the same way as forcing inputs except that one other option appears in the pop-up window.

- Clear Fault - If the device supports latching of Faults, this selection clears a Hardware Output fault indication such as an Off-Wire or Short Circuit condition.

Notes:

A

add devices offline 2-9
add devices online 2-9
add elements to a schematic 5-1
adding DeviceLogix capable devices 3-18
assume data available function 5-7
audience for this manual 1-3

B

bistable function block 4-4
boolean function block 4-2

C

change logic 5-11
change timer and counter values 5-11
comm status override 3-12
common techniques used in manual 1-4
communications menu
- download 3-24
- logic enable off 5-12
- logic enable on 5-12
- upload 3-23

component toolbar 3-29
- discrete input point 3-30
- discrete output point 3-31
- fault input point 3-31
- function block 3-29
- module status point 3-31
- network input point 3-30
- network output point 3-31
- text box 3-30

configuration properties window 5-4
connection options 5-6
- assume data available 5-7
- negate 5-6

connections
- making 5-5

consumed I/O assembly 3-11
counter function block 4-5
- up counter 4-5
- up/down counter 4-6

create a schematic 5-1
- add elements 5-1
- delete a wire 5-6
- make connections 5-5

D

debug 5-12
default I/O information 3-16
delete a wire 5-6
device parameters window 3-4
- comm status override 3-12
- consumed I/O assembly 3-11
- network status override 3-14
- produced I/O assembly 3-10

device properties dialog box 3-1
- device parameters window 3-4
  - comm status override 3-12
  - consumed I/O assembly 3-11
  - network status override 3-14
  - produced I/O assembly 3-10
- DeviceLogix window 3-18
- EDS file window 3-17
- general window 3-2
  - general options 3-3
  - revision field 3-3
- I/O defaults window 3-16

device properties dialog box window buttons 3-3
DeviceLogix
- and RSNetWorx for DeviceNet 1-2
- what is it? 1-1

DeviceLogix and RSNetWorx for DeviceNet 1-2
DeviceLogix components 4-1
DeviceLogix Editor
- menus
  - communications 3-23
    - download 3-24
    - logic enable off 5-12
    - logic enable on 5-12
    - upload 3-23
  - edit 3-21
    - add element 5-1
    - properties 5-4
    - recovery mode 5-4
  - help 3-27
  - tools 3-26
  - view 3-21
- status bar 3-33
- toolbars 3-27
  - component 3-29
    - discrete input point 3-30
    - discrete output point 3-31
    - fault input point 3-31
    - function block 3-29

module status point 3-31
network input point 3-30
network output point 3-31
text box 3-30
online 3-32
standard tools 3-28
Devicelogix Editor
menus 3-20
file 3-21
schematic sheet 3-20
using 3-19
DeviceLogix window 3-18
discrete input point 3-30
discrete output point 3-31
downloading 3-24

E

edit
add element 5-1
edit menu 3-21
add element 5-1
properties 5-4
EDS file window 3-17
EDS files 2-1
register EDS files 2-5
unregister EDS files 2-1
EDS information 3-17
enter device parameters 3-4

F

fault input point 3-31
faults 4-11
file menu 3-21
forcing 5-12
inputs 5-13
outputs 5-13
fucntion blocks
bistable 4-4
boolean 4-2
counter 4-5
up counter 4-5
up/down counter 4-6
timer 4-7
off delay timer 4-9
on delay timer 4-8
pulse timer 4-7
function blocks
description 4-1
inputs and outputs 4-2

G

general window 3-2
general options 3-3

H

hardware inputs 4-10
hardware outputs 4-11
help menu 3-27

I

I/O defaults window 3-16
inputs 4-10
faults 4-11
forcing 5-13
hardware inputs 4-10
network inputs 4-10
status inputs 4-11
interface a DeviceLogix device to a PC 5-10
interface a DeviceLogix device to a scanner 5-8

L

logic enable off 5-12
logic enable on 5-12

M

menus
communications
download 3-24
logic enable off 5-12
logic enable on 5-12
edit 3-21
add element 5-1
properties 5-4
recovery mode 5-4
file 3-21
help 3-27
tools 3-26
view 3-21
menus within DeviceLogix Editor 3-20
module status point 3-31

N

negate function 5-6
network input point 3-30

network inputs 4-10
network output point 3-31
network outputs 4-11
network status override 3-14
node address 3-2

O

off delay timer 4-9
on delay timer 4-8
online operations 5-10
  editing 5-11
    change logic 5-11
    change timer and counter values 5-11
online toolbar 3-32
outputs 4-11
  forcing 5-13
  hardware outputs 4-11
  network outputs 4-11

P

parameter information 3-6
parameters
  understand 3-6
parameters list box 3-10
place and element 5-4
produced I/O assembly 3-10
pulse timer 4-7

R

recovery mode 5-4
register EDS files 2-5
restore default values 3-6
revision field 3-3
Rockwell Automation support 1-5

S

schematic sheet 3-20
select group parameters 3-5
set node address 3-2
standard tools toolbar 3-28
status bar 3-33
status inputs 4-11
support 1-5

T

text box 3-30
timer function block 4-7
  off delay timer 4-9
  on delay timer 4-8
  pulse timer 4-7
toolbars 3-27
  component 3-29
    discrete input point 3-30
    discrete output point 3-31
    fault input point 3-31
    function block 3-29
    module status point 3-31
    network input point 3-30
    network output point 3-31
    text box 3-30
  instruction 3-31
  online 3-32
  standard tools 3-28
tools menu 3-26
troubleshoot and debug 5-12

U

unregister EDS files 2-1
up counter 4-5
up/down counter 4-6
uploading 3-23

W

what we assume about our readers 1-3
window buttons
  device properties dialog box 3-3

Notes:

How Are We Doing?

Your comments on our technical publications will help us serve you better in the future. Thank you for taking the time to provide us feedback.

You can complete this form and mail it back to us, visit us online at www.test.com, or email us at RADocumentComments@ra.rockwell.com

Pub. Title/Type DeviceLogix User Manual

Cat. No. N/A Pub. No. ACIG-UM001A-EN-P Pub. Date June 2001 Part No. 957464-39

Please complete the sections below. Where applicable, rank the feature (1=needs improvement, 2=satisfactory, and 3=outstanding).

Overall Usefulness	1 2 3	How can we make this publication more useful for you?
Completeness (all necessary information is provided)	1 2 3	Can we add more information to help you? ☐ procedure/step ☐ illustration ☐ feature ☐ example ☐ guideline ☐ other ☐ explanation ☐ definition
Technical Accuracy (all provided information is correct)	1 2 3	Can we be more accurate? ☐ text ☐ illustration
Clarity (all provided information is easy to understand)	1 2 3	How can we make things clearer?
Other Comments		You can add additional comments on the back of this form.

Your Name _____ Location/Phone _____

Your Title/Function _____ Would you like us to contact you regarding your comments?

___No, there is no need to contact me

___Yes, please call me

___Yes, please email me at _____

___Yes, please contact me via _____

Return this form to: Allen-Bradley Marketing Communications, 1 Allen-Bradley Dr., Mayfield Hts., OH 44124-9705

Phone: 440-646-3176 Fax: 440-646-3525 Email: RADocumentComments@ra.rockwell.com

PLEASE FASTEN HERE (DO NOT STAPLE)

Other Comments

PLEASE FOLD HERE

PLEASE REMOVE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 18235 CLEVELAND OH

POSTAGE WILL BE PAID BY THE ADDRESSEE

Allen-Bradley
RELIANCE ELECTRIC
DODGE
ROCKWELL SOFTWARE
Rockwell Automation

1 ALLEN-BRADLEY DR
AYFLD

Reach us now at www.rockwellautomation.com

Wherever you need us, Rockwell Automation brings together leading brands in industrial automation including Allen-Bradley controls, Reliance Electric power transmission products, Dodge mechanical power transmission components, and Rockwell Software. Rockwell Automation's unique, flexible approach to helping customers achieve a competitive advantage is supported by thousands of authorized partners, distributors and system integrators around the world.

Americas Headquarters, 1201 South Second Street, Milwaukee, WI 53204, USA, Tel: (1) 414 382-2000, Fax: (1) 414 382-4444
European Headquarters SA/NV, avenue Herrmann Debroux, 46, 1160 Brussels, Belgium, Tel: (32) 2 663 06 00, Fax: (32) 2 663 06 40
Asia Pacific Headquarters, 27/F Citicorp Centre, 18 Whitfield Road, Causeway Bay, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

Publication ACIG-UM001A-EN-P - June 2001

PN 957464-39

© 2001 Rockwell International Corporation. Printed in the U.S.A.